第一篇　化学计量学方法

第一章　现代分析化学与分析信息理论和方法

第一节　现代分析化学——化学量测与化学信息的新兴学科

分析化学学科正经历着巨大的变革[1~4]。由于近年来物理学和电子学的发展，各种新型分析仪器相继问世，昔日的以化学分析为主的经典分析化学已发展成为一门包括众多仪器分析（色谱分析，电化学分析，光化学分析，波谱分析，质谱分析，热分析，放射分析，表面分析，等）为主的现代分析化学。正因为分析手段的不断扩展，广大分析化学家们感到以“溶液平衡”为基础的经典分析化学已不能代表现代分析化学学科发展的全貌，致使 Leihaisky 提出的“不管你喜欢不喜欢，化学正在走出分析化学”的论点广为流传。分析化学界出现了“化学正在走出分析化学”，分析化学应重名为“分析物理”，“分析科学”的议论。1985 年 11 月和1989 年 10 月在维也纳分别召开了第一次和第二次“国际分析化学的哲学和历史会议”，探讨分析化学的某些基本哲学问题。这说明分析化学学科正处在一个急剧分化的发展时期。美国《分析化学》的主编 Murray 在题为“化学量测科学”一文中指出：“用拓展的眼光来看待今天的分析化学是有益和有帮助的，它的发展已使之成为一门创造和应用新概念、新原理和仪器的策略来测量化学体系及其组分的学科，简言之，分析化学已成为一门化学量测科学。”[5]如果我们遵循着“分析化学是一门化学量测科学”的思路，就可以发现，分析化学学科当今的变革不是“化学正在走出分析化学”，而是“物理和新仪器正在走进分析化学”，它使分析化学家手中拥有更多的化学量测工具和手段，为分析化学家解决各学科发展所需解决的复杂的分析难题提供了更有力的武器。如何更有效地使用和发展这些新的分析手段和工具，怎样有效地从这些新的化学量测工具和手段中获取化学家们所需的有关的化学组成和结构信息、以及其他各种有用的化学信息，当是目前分析化学家急需解决的一个新问题。

首先，让我们来考察一下化学量测的全过程。化学量测的全过程似应从选择分析方法和采样开始，经化学量测的试验设计、量测过程的正确控制和优化、分析仪器所得信号的预处理、各类分析仪器数据定性定量分析，再到分析结果的统计推断、分析过程机理研究、所得纯组分波谱图的结构解析等，直至有用决策信息的提取。由此看来，化学量测过程是一个很复杂而且内涵极其丰富的过程，它每一步的有效完成实际都包括了相当丰富的内容，需要有很多关于化学、数学和物理的基础知识。如果说经典的分析化学主要是以“溶液平衡”为基础的话，那么，现代分析化学则是一门包括如何有效地进行各种化学试样的处理（包括分离等化学基础）、各种有关分析仪器所需的物理知识（物理基础）和化学知识（如各种波谱解析）、以及怎样进行最优采样、实验设计或选择最优化学量测方法，并通过解析化学量测数据以最大限度地获取化学及其相关信息（数学基础）的一门综合性极强的化学分支学科。

进行化学量测的基本目的是获取有关物质系统的化学成分与结构方面定性与定量的信息。凯塞尔（Kaiser）在有关分析方法基础的专著中[6]，界定分析化学学科的内涵为取得所研

究的物质组成的知识的有计划的信息过程。卡特曼（Kateman）等[7]从三个方面阐述分析化学的任务：利用已有的分析方法，提供关于物质化学成分的信息——日常例行分析工作；研究利用不同学科的原理、方法取得有关物质系统的相关化学信息的过程——分析化学的科学研究工作；研究利用现有分析方法取得关于物质系统的信息的策略——分析实验的组织工作。柯瓦尔斯基（Kowalski）在一篇题为“分析化学作为信息科学”的论文中[8]，指出分析化学发展史正经历极为重要的时期。这个时期对分析化学学科的重要性，是与现代科学与社会发生的一系列重大变化相关联的。这种重大变化首先源于计算机科学与信息科学的发展。该文援引美国科学基金会的报告，认为信息已成为美国社会的极重要组成部分，约半数的劳动力从事与信息相关的工作，挣取一半以上的劳动收入。在这样一个信息具有如此重要地位的社会中，分析化学起什么作用？该文作者认为，分析化学现在是，而且过去也一直是一门信息科学。在化学的各个分支学科中，分析化学担负的任务与其他分支学科的不同处，就在于分析化学的研究对象，不是直接提供某种具体的例如无机和有机材料，而是提供与这些材料的化学成分和结构相关的信息，研究获取这些信息的最优方法与策略。当然，分析化学工作者是与其他化学工作者分工而又合作，共同去完成生产与科研向化学提出的使命的。不但分析化学与无机化学、有机化学等传统的化学分支学科关系如此，分析化学与一些新兴的边缘学科如环境化学的关系亦是如此。例如，在IUPAC30届学术大会上关于“环境的挑战”的学术讨论中，出现这样的学术报告题目：“分析数据的生物学意义——所有环境课题都源于分析化学家！”[9]。将分析化学认作通过化学量测取得化学信息的科学，并不是说分析化学发展到今天才具有这种性质，也并不完全是由于信息对当代社会的重要性，人们才有意强调这一事实。然而，在分析化学得到飞速发展的今天，重新认识分析化学作为通过化学量测来提供化学信息的科学这一性质，反映了分析化学的新发展，而且，这一新发展可能还是质的飞跃。那就是，分析化学工作者已不仅是单纯的分析数据的提供者，而是解决实际问题有用的化学信息的提供者。

值得提出的是，近年来随着计算机，特别是微型计算机大量进入化学特别是分析化学实验室，一门新的化学分支学科——化学计量学应运而生，为有效进行化学量测和提供化学信息开辟了新通路，为分析化学提供了新机遇。化学计量学是一门交叉学科，它应用数学、统计学与计算机科学的工具和手段及其最新成果来设计或选择最优化学量测方法，并通过解析化学量测数据以最大限度地获取化学及其相关信息，自然，它首先就在分析化学中得到了普通认同。自20世纪70年代中期诞生以来，在20世纪80年代得到长足发展，至今已逐步日趋成熟。图1-1将化学计量学与分析化学作为一门化学量测和化学信息学科之间的关系以图示方法表示出来了。

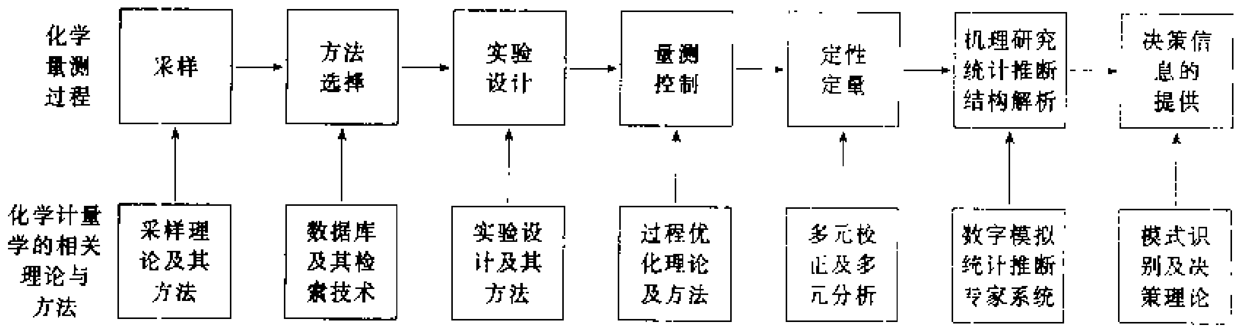


图1-1　化学量测流程与化学计量学方法的关系示意图

从此图我们可以看出，对于化学量测的每一步，化学计量学都有相应的理论和方法学，研

究如何来使化学量测和数据解析过程变得更有效，实质上，化学计量学就是一门关于化学量测的理论基础和方法学的化学分支学科[10]。分析化学与化学计量学有着如此密切的关系，就不难理解为什么美国化学会的分析化学杂志——“Analytical Chemistry”要将化学计量学作为一个分支领域，从1980年起就每两年对它的发展进行一次总结性综述了。我国分析化学家高鸿先生在十几年前就预言过：“数学在分析化学中的应用日益重要，如果说20世纪60年代是分析化学与电子学结合的时代，70年代是分析化学与电子计算机结合的年代，80年代很可能是分析化学与统计学和应用数学紧密结合的时代”[11]。从这一角度来看，化学计量学已经成为分析化学学科的必要数学基础。我们将化学计量学中从分析化学的信息理论基础、化学采样理论及方法、化学量测过程的试验设计与优化、化学量测数据的多元校正和多元分辨的定性定量解析，一直到分析过程机理研究、数值模拟和计算机波谱解析、有用决策信息的提取，其中包括化学模式识别、化学构效关系研究直至人工智能与化学专家系统的所有内容汇成一书，来作为分析化学手册的第十分册，其初衷也在于此。

分析化学作为一门化学量测和化学信息科学，对其量测过程有效性及效率的估计和评价就显得十分重要了。分析信息理论从信息理论的角度来研究化学量测过程。如果我们将通讯处理信息的过程与化学分析中的化学量测过程来进行比较，就可容易发现这两个过程十分相似（参见图1-2）[12]。

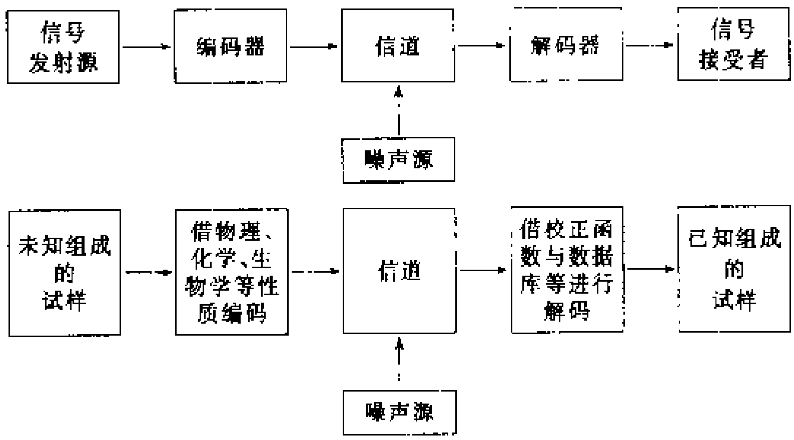


图1-2　通讯信息处理过程与化学量测过程的比较

进行化学量测的目的就在于消除或减少被量测的化学系统在化学成分、结构及其他相关信息的“不确定度”。分析信息理论和方法可提供相应的概念和方法来定量表征化学量测系统的“不确定度”及化学量测过程中体系之“不确定度”的消除或减少的定量度量（或称为化学量测过程的信息量之获取）。它不但可以对不同的分析过程，如分离、定性鉴定、定量测定等进行信息理论分析，还可对现代分析化学中各类复杂分析仪器的提供信息之能力进行合理评价，为分析化学量测提供了一套选择不同分析过程或不同分析仪器的理论基础和具体定量方法。本章将系统地介绍这些概念和方法。

第二节　熵——化学量测的“不确定度”的定量度量

一、分析试验与不确定度

分析工作的目的是取得有关未知试样的化学成分与结构的相关信息。所以，在进行分析测试之前，必存在某种不确定性或称“不确定度”。设有一份试样，其中可能含有k种离子中

的某一种，定性分析的任务就是确定这一离子确切是何种离子。在分析测试之前，存在有 k 种可能性，或称分析实验有 k 个可能的结局。如将上述情况用数学式表示，并设上述 k 种可能结局为 a_1，a_2，…，a_k；发生这 k 种可能结局的概率为 p_1，p_2，…，p_k，则有

$$A=\begin{matrix} a_1, & a_2, & \cdots, & a_k \\ p_1, & p_2, & \cdots, & p_k \end{matrix}$$

在此 A 表示发生上述定性分析实验的一个事件。如果存在有两个不同定性分析实验的事件，分别记为 A_1 和 A_2，并由以下的形式表示出的两个数表表示出，即

$$A_1=\begin{matrix} a_1, a_2, \cdots, a_k \\ 1, 0, \cdots, 0 \end{matrix}; \quad A_2=\begin{matrix} a_1, & a_2, & \cdots, & a_k \\ 1/k, & 1/k, & \cdots, & 1/k \end{matrix}$$

很明显，因为对于事件 A_1，它发生的可能性实际上只有一种，即只可能发生 a_1，而发生 a_2，…，a_k 的概率都等于零，所以，此事件的"不确定度"实际上不存在，如果存在这样一种"不确定度"的定量度量标准，对于事件 A_1，其值应为零；然而，对于事件 A_2，它发生的可能性就有 k 种，即发生 a_1，a_2，…，a_k 的概率都相等，故此事件的"不确定度"很大，如存在一种"不确定度"的度量标准，那么对于事件 A_2，其值应该很大或至少要大于零。以下我们将要讨论的熵的概念，就是这样"不确定度"的一种定量度量。

二、不确定度与仙农熵

对于一个随机事件的数表，仙农熵的定义如下

$$H=-\sum_{i=1}^{k} p_i \log p_i \tag{1-1}$$

仙农给出了上述定义并称 H 为熵。lg 一般表示以 e 为底，此时所得熵的单位为奈特（nat）；如果以 10 为底，其单位为的特（dit）；而以 2 为底时，其单位则为比特（bit）。从式（1-1）可知，如对前述的两个实例，用式（1-1）计算可得

$$H(A_1)=-1\log 1=0$$

$$H(A_2)=-\sum(1/k)\log(1/k)$$

$$=-\lg(1/k)=\log(k)$$

从上述结果明显可知，事件 A_2 的不确定度大于事件 A_1 的不确定度。

三、仙农熵的性质

（1）非负性，即

$$H(p_1,p_2,\cdots,p_k)\geqslant 0$$

这是因为 $p_i(i=1,2,\cdots,k)$ 为概率，故有 $1\geqslant p_i\geqslant 0$，所以 $\lg(p_i)$ 不可能取正值，即 $(-\sum p_i\log p_i)$ 才取正值。

（2）对于确定性事件，其熵为零。即

$$H(1,0,\cdots,0)=0$$

所谓确定性事件，即为该事件只有一种可能结果，如用数学语言表达则为，对于发生某一可能结果的概率为 1，而发生其他可能结果的概率都为零。前面讨论的事件 A_1 就是确定性事件的一个例子。注意到在此实际上是引入了一个人为的假设，即

$$\lim_{p\to 0}(-p\log p)=0 \tag{1-2}$$

式中，p 为一表示概率的实数。

引入这一假设是因为 log (0) 在数学上为一无意义的数。可是，如从极限的角度来看，引入 (1-2) 式在数学上是完全合理的。如假设 $p=1/e^n$，此时

$$-\log p=n$$

而

$$-p\log p=n/e^n\to 0 \tag{1-3}$$

随着 n 的增大，n/e^n 将很快接近于零。实因此时 p 是一个比 ($-\log p$) 更高阶的无穷小。所以上式实际上是用极限的概念来避免对 log(0)的直接计算。有了式 (1-3) 的定义，所以有

$$H(1,0,\cdots,0)=-1\log(1)-\sum 0\log 0=-1\log(1)=0$$

(3) 对于等概率结果的事件，其熵最大。即

$$H(p_1,p_2,\cdots,p_k)\leqslant H(1/k,1/k,\cdots,1/k)=\log(k) \tag{1-4}$$

所谓等概率结果的事件，即该事件有 k 种可能结果，而且，发生这 k 种结果的概率都相等，即发生某一可能结果的概率都为$(1/k)$。由此性质可知，等概率结果的事件的不确定度最大。

四、条件熵与可疑度

今考察两个随机试验 A 与 B，其可能结果为

$$A:\ A_1,\ A_2,\ \cdots,\ A_n$$

$$B:\ B_1,\ B_2,\ \cdots,\ B_m$$

则两事件同时发生时的熵为

$$H_{n,m}(AB)=-\sum_{i=1}^{n}\sum_{j=1}^{m}p(A_iB_j)\log[p(A_iB_j)] \tag{1-5}$$

条件熵的定义：如果试验 A 出现结果 A_k 时，试验 B 在此条件下的熵为

$$H_m(B|A_k)=-\sum p(B_i|A_k)\log[p(B_i|A_k)]$$

相对应也可定义试验 B 出现结果 B_k 时，试验 A 在此条件下的熵

$$H_n(A|B_k)=-\sum p(A_i|B_k)\log[p(A_i|B_k)]$$

更进一步，还可定义在进行试验 A 的前提下试验 B 的条件熵

$$\begin{aligned}H(B|A)&=\sum p(A_k)H_m(B|A_k)\\&=\sum p(A_k)[-\sum p(B_i|A_k)\log[p(B_i|A_k)]]\\&=-\sum\sum p(A_k)p(B_i|A_k)\log[p(B_i|A_k)]\end{aligned}$$

或在进行试验 B 的前提下试验 A 的条件熵

$$H(A|B)=-\sum\sum p(B_k)p(A_i|B_k)\log[p(A_i|B_k)] \tag{1-6}$$

条件熵的性质：

① $H(A|B)=H_{n,m}(AB)-H_m(B)$

$H(B|A)=H_{n,m}(AB)-H_n(A)$

此性质可由积事件(即同时发生的事件)的概率公式(参阅第十一章)直接导出。

② $H(A|B)\leqslant H_n(A)$

$H(B|A)\leqslant H_m(B)$

此性质的证明可参阅文献 [12]。

③ 如果随机试验 A 与 B 相互独立，此时 $p(A|B)=p(A)$(参阅第十一章)，亦即分析实验 B 与待解决的分析课题 (试验 A) 毫不相干，当然做试验 B 将得不到试验 A 的任何信息，也无从减少关于试验 A 的“不确定度”。此时，有

$$H_{n,m}(AB)=H_n(A)+H_m(B) \tag{1-7}$$

五、可疑度、互信息与散度

可疑度的定义：从条件熵的定义可知，$H(A|B)$实际表述了在进行了试验B以后的试验A的“不确定度”。所以，$H(A|B)$又称为可疑度。这样定义的可疑度，实际上可以说是反映了分析仪器或方法提供信息的能力与解决给定分析课题的需要之间的差距的定量度量。

互信息的定义：

$$I(A;B)=H_n(A)-H_n(A|B)=H_m(B)-H_m(B|A)=I(B;A) \tag{1-8}$$

式中，$I(A;B)$称为B关于A的互信息，或$I(B;A)$称为A关于B的互信息。同时，式(1-6)也可称为信息守恒定律或信息平衡原理。

散度或卡尔贝克(Kullback)信息量的定义：

设有p、q两个概率分布，

验前分布 $p:(p_1,p_2,\cdots,p_n),\sum p_i=1$

验后分布 $q:(q_1,q_2,\cdots,q_n),\sum q_i=1$

当分布p被分布q取代时，散度或卡尔贝克信息量定义为

$$I(q /\!/ p)=\sum q_i\log(q_i/p_i) \tag{1-9}$$

类似地可给出连续变量的散度或卡尔贝克信息量的定义：

$$I(q /\!/ p)=\int q(x)\log(q(x)/p(x))\mathrm{d}x \tag{1-10}$$

第三节　定性分析的信息理论和方法

定性分析作为分析工作的重要组成部分，提供的是关于物质成分、结构特征方面的化学信息，回答的是“是什么?”这一问题。在这一节我们将就有关定性鉴定方法的信息定量评价方法、色谱及层析方法实验调优的信息理论和方法、质谱及红外光谱的编码与检索的信息理论和方法等方面来分别加以介绍。

一、不同定性分析鉴定方法的信息量估价

在定性分析鉴定方法的信息量估价中，一般可分为两种不同的方法，一种是针对一具体定性实验而言，以实验前的结果“不确定度”与实验后的结果“不确定度”之差别来估价此定性实验的信息量，亦即用前节所讨论的实验前与实验后的熵之差来估价实验的信息量，这样的信息量估价可包括以下几种情况：

(一) 结构定性分析的信息量

关于结构分析的结果之信息量计算十分简单。结构分析过程可用图1-3所示的简单框图表出。

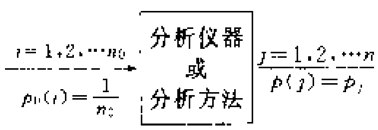


图1-3　结构分析过程单框示意图

作为分析仪器或方法的输入，可以是n_0个等概率的可能化学结构。而分析仪器或分析方法的输出，可能给出n种尚不能分辨的结构。每种结构的验后概率一般是不相等的。结构分

析提供的信息量可按下式来进行计算。

$$I(A/\!/B)=H(A)-H(B)$$

在此，A 表示试验前可能存在不同化学结构的事件；B 表示进行了分析仪器或分析方法试验后可能存在不同化学结构的事件；而 $I(A/\!/B)$ 则表示在进行了分析仪器或分析方法的试验 B 后所获得的信息量。很明显，进行了分析仪器或分析方法的试验 B 后所获得的信息量应为 A、B 两事件的“不确定度”即熵之差[13]。

分析前：

$$p_0(i)=1/n_0 \qquad (i=1,2,\cdots,n_0)$$

$$H(p_0)=\ln(n_0)$$

分析后：

$$p(j)=p_j \qquad (j=1,2,\cdots,n)$$

$$H(p)=-\sum_{j=1}^{n}p_j\ln(p_j)$$

$$\begin{aligned}I(A/\!/B)&=H(A)-H(B)\\&=\ln(n_0)+\sum p_j\ln(p_j)\end{aligned}$$

如假定实验后各可能结构均为等概率，则

$$I(A/\!/B)=\ln(n_0/n) \tag{1-11}$$

注意到此时 $I(A/\!/B)$ 与卡尔贝克（Kullback）信息量定义完全一致了。

（二）定性化学反应分析的信息量

设有一纯溶液，它可能是下述离子之一的试液：Ag^{+}，Pb^{2+}，Al^{3+}，Zn^{2+}，Na^{+}或K^{+}。今用经典定性分析方法进行分离鉴定。设加入试剂盐酸，如何估算正反应（发生沉淀）及无反应时的信息量[14]？

正反应（发生沉淀）时，因可与盐酸产生白色沉淀的在这 6 种离子中只可能是 Ag^{+}和 Pb^{2+}，所以，经此反应后，可能离子的范围从 6 种变成了 2 种，根据式（1-11），可得其化学反应过程所得的信息量为

$$I(A/\!/B)=H(A)-H(B)=\ln(6/2)=\ln3$$

无反应时，因不与盐酸产生白色沉淀的在这 6 种离子中可能为 Al^{3+}，Zn^{2+}，Na^{+}或 K^{+}中任何一种，所以，经此反应后，可能离子的范围从 6 种变成了 4 种，根据式（1-11），其化学反应过程所得信息量为

$$I(A/\!/B)=\ln(6/4)=\ln(1.5)$$

（三）测定物理常数鉴定有机化合物

测定熔点、沸点、折射率、密度等物理性质常用于有机化合物的鉴定。今试以纯物质的熔点测定为例，来考察某次物理常数测定所获得的信息量。设所测物质在分析之前就已知应是属于在温度 100℃ 至 200℃ 的一种物质，又已知在此温度范围之内可能存在 200 种有机化合物，如进而设它们将等概率分布于此温度区间，则在量测之前的不确定度，即熵为

$$H(A)=\ln(200)$$

经量测后知，其熔点为(100±1)℃，从已知在此温度范围之内可能存在 200 种有机化合物且它们将等概率分布于此温度区间，则测定后的熵为

$$H(B)=\ln[(200/100)\times2]=\ln(4)$$

此实验的信息量为

$$H(A)-H(B)=\ln(200/4)=\ln(50)$$

二、仪器定性分析的信息量

如前所述，在定性分析的信息量估价中，一般可分为两种不同的方法，前一节已经讨论了针对某一具体定性实验信息量估价方法，亦即用实验前与实验后的熵之差来估价实验的信息量，在这一节里，将讨论另一种专门针对仪器定性分析的信息量估价方法，这类方法之主要思路是直接估价仪器信号的熵，亦即该仪器可获得的信息量。化学计量学方法就是通过采用化学实验的方法努力增大仪器的分析能给出的信息量或是采用编码的方式使得仪器分析的结果能尽量多地给出信息，以达到用信息量来作为目标函数，从而提高仪器定性分析的效率。

（一）薄层及纸上层析分离与定性鉴定

薄层色谱与纸色谱分离一般以 R_f 值表述，不同的化学物质具有不同的 R_f 值。如将薄层色谱分成 m 个间距，对落入每个间距的 R_f 值的个数 n_k 计数，则薄层及纸上层析分离与定性鉴定的信息量可由下式给出[15]：

$$I=-\sum_{k=1}^{m}(n_k/n)\ln(n_k/n) \tag{1-12}$$

此式是较早将信息理论引入薄板层析的马萨特（Massart）[15]所用的式子，它的物理意义是很直观的。如 n 个化合物均落在某个间距中，则此时 I 为零，未获得如何分离与信息。I 值只有在 n 个化合物均匀分布在 m 个间距时最大。此信息量可用作寻找最佳展开试剂的目标函数。

（二）色谱分离鉴定的信息量

在色谱分析中，一般是利用保留指数来进行定性分析．与薄层色谱和纸色谱的不同之处是，每一个化合物都由一个色谱峰来表示，存在着色谱峰相互重叠的问题。所以，色谱分离鉴定的信息量比起薄层色谱和纸色谱的信息量计算多了一项．仿薄层色谱的处理方法，将保留时间进行区间离散化，即划分为等长的（Δy）m 段，统计一定数量的化合物（总数 n）的保留指数落入不同段的频数，则有

$$I=-\sum_{k=1}^{m}(n_k/n)\ln(n_k/n)+\ln(\Delta y)-\ln\sqrt{2\pi e\sigma_c^2} \tag{1-13}$$

式中，$\ln(\Delta y)$ 为一个常数，所以，对于不同的色谱柱，此项没有差别，一般可采用将 Δy 取为 1 而去掉。式中的第三项 $\ln\sqrt{2\pi e\sigma_c^2}$ 实际是来自前一节所定义的可疑度，在此表述了在进行了色谱分离以后的化合物靠保留指数来定性时的“不确定度”，也就是说是色谱峰的熵。其中 σ_c 为色谱峰的标准差。注意到在此我们是假设了每一种物质的色谱峰的标准差都是一样的。因为由式（1-6）定义的可疑度为

$$H(A|B)=-\sum\sum p(B_k)p(A_i|B_k)\log[p(A_i|B_k)]$$

在此，$p(B_k)$表示第 k 种化合物出现的概率，在此假设为等概率，即（$1/n$），$p(A_i|B_k)$ 为第 k 种化合物在色谱仪中的信号，一般假设为标准差为 σ_c 高斯色谱峰。对此积分所得结果就是 $-\ln\sqrt{2\pi e\sigma_c^2}$。详细讨论可参见文献［12］。由式（1-12）定义的信息量标准，可用于选择不同流动相或色谱柱。

（三）质谱定性鉴定的信息原理

质谱是本世纪出现的分析方法。质谱仪的功能是产生带电离子，包括母离子和原分子的离子碎片，并按离子的质荷比（m/z）对化合物进行区分，“质谱”是不同离子数目的记录，每种离子的相对数目对每种化合物（包括同分异构体）将是特征的。质谱仪能提供关于有机化合物结构和固态试样元素分析的大量信息。所谓“化合物的质谱”，包含大量的离子碎片，且这些碎片离子的相对丰度时常超过母离子。分子碎片化的独特性有助于化合物的鉴别工作。质谱鉴定可用不同的方法进行。一种是解析法，即研究与假设破碎模式，并反过来从纯化合物质谱中的碎片离子来构思分子的结构。另一种办法是检索，即不管质谱图的含义，从已有的数据库中检索。本小节主要讨论信息理论在质谱检索中的应用。

利用质谱进行定性鉴定，我们先对其讯号能提供的信息量作一粗略估计。设用一低分辨率质谱仪，其质荷比区间仅 200 原子质量单位，如用信息论的观点就是 200 个信道。又设每一质量数位置我们仅区别有峰（其编码为 1）与无峰（其编码为 0），使用这种 0-1 编码时，则每个信道的最大可提供的信息量为 1bit（ld2，在此 ld 表示以 2 为底的对数），如果做到每个信道都相互不相关，这一低分辨率的质谱仪理论上可提供 200bit 的信息量，即大约能分辨 2^{200}种不同化合物，这个数目当然远远大于目前已知的有机化合物的个数。

现试从统计角度来探讨一下将谱图编码后、互相重复的可能性。设有 N 个谱图，按 0-1 编码，每个谱图可认为含 n 个“信道”，即原子离子质量单位数。在研究谱图 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 的重复问题时，可将每一谱图认作一个向量：

$$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$$

$$\boldsymbol{y}=(y_1,y_2,\cdots,y_n)$$

向量中的每一元素 x_i 或 y_i 的取值只有 0 或 1，即有峰时为 1，无峰时为 0。定义函数 $F(x_i,y_i)$，将编码重复的情况及不重复的情况可分别表述为

$$F(x_i,y_i)=0 \qquad 当\ x_i=y_i$$

$$F(x_i,y_i)=1 \qquad 当\ x_i\neq y_i$$

再定义函数 D，以表述所有信道中编码不重复数之和，

$$D=\sum_{i=1}^{n}F(x_i,y_i)$$

可以证明函数 D 的期望值 D 在信道 i（在质谱中即为某一质荷比数）出现 1 的概率，及信道 i 出现 0 的概率皆为 0.5 时可达最大[12]。

实际上，如从信息理论来考虑，同样可以得出上述结论。对于质谱的 0-1 编码，每个信道 i 的熵值可由式（1-14）表出：

$$H_i=-\sum_{k=1}^{2}p_{ik}\mathrm{ld}\,p_{ik}=1 \tag{1-14}$$

式中，p_{ik}（$k=1$，2）表示信道 i 在 N 张质谱图中出现 1（$k=1$ 时）或 0（$k=2$ 时）的概率，显然，根据等概率事件熵最大原理，只有当 $p_{ik}=0.5(k=1,2)$时，H_i 为最大。当质谱的 n 个信道都相互不相关时，此时对 N 个质谱的 0-1 编码的总信息量为

$$H=\sum_{i=1}^{n}H_i=\sum_{i=1}^{n}\left[-\sum_{k=1}^{2}p_{ik}\lg p_{ik}\right]=n$$

但实际编码由于各信道之间存在相关性，上述所得实际是 0-1 编码可以得到的极大值，将其记为 $H_{\max}$。定义实际可得信息量 H_n 与最大可得编码信息量 $H_{\max}$之比为编码效率(code efficien-

cy, CE),

$$CE\% = (H_s/H_{max}) \times 100\% = (H_s/n) \times 100\%$$

在 20 世纪 70 年代初期关于信息原理用于质谱编码检索的早期论文中[16,17]，研究了如何进行编码能提供最大的信息，而又尽可能节省计算机内存与缩短检索时间等问题。

在 0-1 编码的问题中，需确定在何种情况下应定义信道 i 具有峰即编码为 1。为此需确定一临界水平值(TL 值)，强度大于(或等于)此值认作有峰，否则认作无峰。显然，TL 值如选择过低，则噪声信号亦可能被认作分析信号，在极端情况下，每一信道均出现 1，信息量将降至零；如 TL 值取得过高，则许多微弱强度的分析信号均被认作为零，亦将损失信息量，以基峰的 0.01%至 0.3%强度作 TL 值时，信息量是恒定的，当 TL 值上升到约 1%以上时，信息量下降。因取很低的 TL 值易受噪声的影响，折衷的方案是取 1%的 TL 值。当然，对不同的质量数取不同的 TL 值，或通过调整 TL 值使相应质量数的信息量 H_i 达到最佳值。

TL 值对信息量有一定影响，但借调整 TL 值增大信息量是有限的。p_{ik}值基本上是质谱峰的自然分布概率所决定的。有的质量数位置出现峰的概率很小，有的 p_{ik}值则较大。减少对计算机内存要求及增大编码效率的办法之一，是舍弃 p_{ik}差别很大的质量数位置，这对信息量影响不大，但能减少对内存的要求。更好的办法是在考虑到信道之间相关性的基础上，将某些信道合并，以尽量减少信道而使信息量减少不多。如已知质量数为 86 的 $p(86_1)$(出现有峰的概率)为 0.236，而质量数为 87 的 $p(87_1)$为 0.246，如将二信道合并，在此两信道信号不相关的条件下，可得 $p_{合,1}$为 0.48。然而，实际上并非如此。对数据库进行分析可知，如在质量数为 86 位置出现峰，则有 56%的情况在质量数为 87 处亦有峰，即条件概率 $p(87_1|86_1)=0.56$，可见两质量数位置(此两信道)是相关的。下就以此例来说明二信道合并时信息量的计算。注意到两信道合并后，所谓有峰(编码为 1)的情况包括以下三种情况：① 在质量数为 86 位置出现峰，且在质量数为 87 处亦有峰；② 在质量数为 86 位置出现峰，但在质量数为 87 处无峰；③ 在质量数为 86 位置无峰，但在质量数为 87 处有峰。所以，出现编码为 1 的概率应为

$$\begin{aligned}
p_{合,1} &= p(87_1,86_0)+p(87_0,86_1)+p(87_1,86_1)\\
&= p(87_1)p(86_0|87_1)+p(86_1)p(87_0|86_1)+p(86_1)p(87_1|86_1)\\
&= p(87_1)[1-p(86_1|87_1)]+p(86_1)[1-p(87_1|86_1)]+0.246\times0.56\\
&= p(87_1)\{1-[p(86_1)p(87_1|86_1)]/p(87_1)\}+0.236[1-0.56]+0.246\times0.56\\
&= p(87_1)-[p(86_1)p(87_1|86_1)]+0.236[1-0.56]+0.246\times0.56\\
&= 0.246+0.236-0.236\times0.56\\
&= 0.35<0.48
\end{aligned}$$

可见，上例合并二峰位置得不到出现 1 的概率 0.48。对质谱峰的研究表明，二质量数相差 1 (1H)、2(2H)、13(CH)、14(CH_2)、15(CH_3)等数值时，相关关系最为显著。相关关系本身在编码中可考虑作为压缩维数以节约内存和缩短检索时间的依据。显然，将相关的质量数位置合并损失有用信息较少。

Wangen 等[17]对有关文献编辑的质谱图集[18]中 6652 条低分辨质谱图进行编码试验。这些质谱系在不同实验室用不同的电子轰击离子化仪器(30～100eV)测得，涉及约 5000 种不同的化合物，即不少化合物在此图集中存在不同实验室测得的互相重复的质谱。借此可校验编码的使用效果。作者比较了 0.1%、1.0%、5.0%三种 TL 值用于编码，约 5000 种化合物摄制的 6652 条质谱经编码后，相互重复的组数见表 1-1。由表可见，TL 值由 0.1 升至 1.0%，相似异构体或相似化合物给出重复的质谱编码的情况并无显著变化，1.0%是较佳的 TL 值。

表 1-1　编码质谱的相互重复组数

TL 值/%	重复组数	涉及谱图数	结构重复的化合物特征		
			同一化合物	相似异构体	相似结构及其他
0.1	167	370	112	42	14
1.0	209	451	156	38	15
5.0	376	836	335	98	43

表 1-1 涉及 352 个质量位置。按上述合并质量数位置的办法，将原 352 个质量数合并至 80 个，并取 *TL* 值为 1%，在表 1-1 的基础上增加的重复编码数见表 1-2。如利用相关关系进一步合并一些质量数位置，使信道数压缩至 48，进一步增加的重复编码质谱图数目亦列于表 1-2。从表 1-2 可以看出，经压缩维数后信道的编码效率 *CE* 值增高。80 信道的最大熵值为 80bit，实际提供的信息量为 76.5bit，故编码效率为 95.6%，较使用 352 维信道时，虽绝对信息量由 131.7bit 降至 76.5bit，编码效率则较原 37.4%提高甚多，即能更有效地利用有限的计算机内存。

表 1-2　压缩编码维数对检索的影响

维　数		80	48
较维数为 352 时增加的重复质谱图数		51	230
增加的重复谱	同一化合物	22	76
图涉及的化合	类似异构体	13	115
物特征	相近结构及其他	16	60
新增重复编码所涉及的谱图数		115	555
总信息量/bit		76.5	45.8
编码效率(*CE*)		95.6	95.4
与 352 维编码比较减少的熵值/bit		55.2	85.9

上述讨论的 0-1 编码是最简单的编码。实际上，在某一质量数位置有峰时，峰的强度还包含有化学信息。设每一信道的峰强度可分成 m 个阶梯或水平，在编码时应将这部分信息亦加以考虑。令 p_{ik} 为 i 信道的峰落入 k 水平的概率，则在各信道是相互独立的情况下，有

$$H_n = \sum_{i=1}^{n} H_i$$

其中

$$H_i = -\sum_{k=1}^{m} p_{ik} \lg p_{ik}$$

（四）红外光谱定性鉴定的信息原理

上节讨论的关于质谱编码的信息原理，对各种电磁波谱分析原则上均是适用的。各种电磁波谱中，红外光谱是较常用的有机化合物波谱鉴定手段。本节将以红外光谱为例进行简要讨论，其他电磁波谱分析可举一反三类推，如将前节中的质量数位置改为红外光谱的波数或波长小区间，可对红外吸收光谱进行编码，凡吸收等于或超过某一约定的临界水平 *TL* 值者均编码为 1，否则为 0，即得 0-1 编码，如再将吸收峰带强度划分为若干水平，则还可进一步得到更多的化学信息。

有关文献[19]按上述方法对 ASTM 红外光谱索引（ASTM Infrared Spectral Index）中的红

外光谱数据库进行编码，所有文件包括了约102000条红外光谱（按Wy-andotte-ASTM法编码[20]是将2.0～15.9μm波长区域划分为宽度为0.1μm的140波长段，在每一波长段，如吸收强度超出选定的TL值，则编码为1，否则为0）。由ASTM文件编码96900条红外光谱（WL-1）。如无误差及相关关系的影响存在，WL-1文件的信息量在选取100个峰时约50bit。实际由于误差及相关关系的影响，只能达到20bit左右。当然，20bit信息量仍足以区别10^6个化合物，故其信息功能仍是很强的。红外光谱与质谱有不同处，因波长区段或波数区段的划分与质量数相比，有任意性。为了说明波长（或波数）区段划分的宽度对信息量的影响，取WL-1文件的一个子集WLS-1（含5100条光谱）进行扩充谱带"窗"的试验。按0.1μm"窗"宽度编码后，对含有峰（编码为1）的波长段作如下处理：如某一波长区段含有峰（1），则将其相邻的二波长段亦赋予"1"，即得"窗"宽为0.3μm的编码（记为WLS-1-3）。进而将WLS-1-3文件再按相同办法扩"窗"，可得"窗"宽为0.5μm文件（WLS-1-5）。这一扩窗步骤示于表1-3，对原按0.1μm波长区段编码的文件进行上述处理后，对信息量将发生影响。这种影响与三个因素有关：① 扩"窗"后由于许多波长区段的"1"增加，即概率p_1值增加，信息量一般上升。当然，p_1值增至超过0.5以后信息量又下降。这个因素可称为概率效应。② 误差效应。误差将使信息量降低。随着"窗"扩大这一影响的相对值将减弱。③ 相关效应。扩"窗"肯定导致相关关系的增加。例如j位置有峰，$j-1$与$j+1$亦赋值"1"。这导致信息量降低。表1-4例举将WLS-1的窗扩至0.3μm(WLS-1-3)及0.5μm(WLS-1-5)信息量较WLS-1发生的变化。"+"代表信息量增加，"－"代表减少。对WLS-1本身而言，亦存在误差与相关关系对信息量的影响，在表中亦列出以资比较。总之，将光谱编码的"窗"适当扩充，例如由0.1μm扩至0.3μm，对增加信息量是有利的。

表1-3　红外光谱编码的扩"窗"步骤示例

文　件	"窗"宽/μm	峰位置						
		$j-2$	$j-1$	j	$j+1$	$j+2$	$j+3$	$j+4$
WLS-1	0.1	0	0	1	0	1	0	0
WLS-1-3	0.3	0	1	1	1	1	1	0
WLS-1-5	0.5	1	1	1	1	1	1	1

表1-4　红外光谱编码信息量（bit）与"窗"宽的关系

影响因素	WLS-1	WLS-1-3	WLS-1-5
概率效应（＋）		22.6	8.3
误差效应（－）	26.5	12.3	6.1
相关效应（－）	6.1	20.7	14.0
总效应		－8.0	＋0.4

对于WL-1文件，当取全部140信道编码时，其编码效率为15%。择优选出93信道的编码效率增至23%。如无误差存在，140信道的效率当为41%，93信道当为51%。可见，误差效应对信息量有重大影响。扩"窗"能提高编码效率。例如，对优选的信道，当WLS-1的编码效率为24%时，扩至WLS-1-3为41%，WLS-1-5为46%。

在记录用于编码检索的红外光谱时，应固定实验操作条件，例如先将整个光谱区的基线调至透射率（T）＝（95±2)%，继将最强的谱带调整至T＝（5±2)%（改变吸收池厚度），不时校正波长表度，对所用化合物进行严格提纯精制，峰强度取最高峰处的透射率与基线的差值：峰强度低于一定TL值时编码为0，否则为1。TL值较高时，出现的"0"增加，使有

用的可供选用的信道数下降。文献［21］采用了3%、5%、10%三种 TL 进行比较，不同 TL 值选出的有用信道数及信息量列于表1-5和表1-6。表1-5是将波数（wave number，WN）由 $4000cm^{-1}$ 按每 $25cm^{-1}$ 为一段均分为140段（称WN-码），表1-6是将波长（wave length，WL）2.0μm起按每0.1μm波长段分为140段（称WL-码）。误差项的影响与 TL 值有关，在将最强带调整至5%时，若 $T=1\%$ 的误差，则在近 TL 值处造成的误差当 TL 为3%时 $T=0.25\%$，当 TL 为10%时 $T=0.7\%$，作较坏估计，取 TL 值为3%时，由此造成的信息量下降为0.03bit。另一项可能的误差是记录峰值的实验误差，对WL码，在长波长区这一误差将较显著，故WN-码的可靠性较高。光谱区间划分按WL-码在（3500～2500）cm^{-1} 区域峰数目较少，这是由于在这一区域将两个或两个以上的峰编码为一个“1”，用WN-码则可免此弊，故WL-码的信息量一般很低。对较高波数区，则可能出现相反情况，但考虑到相关关系等因素，总的说来WN-码的信息量较大，故选择WN-码一般效果较佳。

表1-5 TL 值对WN-码信息量（bit）的影响

数据集①	$TL=3\%$		$TL=5\%$		$TL=10\%$	
	信息量	信道数	信息量	信道数	信息量	信道数
CHS-WN	15.1	26	13.4	22	7.5	11
CHU-WN	31.5	47	29.7	42	24.9	35
CH-WN	32.8	58	18.8	19	25.0	47
ALC-WN	28.7	45	28.7	44	28.5	45
ETH-WN	17.5	21	15.0	17	22.2	34
CARB-WN	15.5	28	15.4	26	14.9	25
A-WN	41.3	73	39.9	65	36.5	61

① CHS——饱和烃；CHU——不饱和烃；CH——饱和烃及不饱和烃；ALC——醇；ETH——醚；CARB——醛/酮；A——以上全部谱图。

表1-6 TL 值对WL-码信息量（bit）的影响

数据集	$TL=3\%$		$TL=5\%$		$TL=10\%$	
	信息量	信道数	信息量	信道数	信息量	信道数
CHS-WL	12.2	25	9.6	18	5.8	18
CHU-WL	26.3	15	25.5	50	19.9	34
CH-WL	28.2	57	26.1	50	16.9	32
ALC-WL	24.8	48	24.5	48	24.1	48
ETH-WL	21.2	35	20.5	36	20.1	36
CARB-WL	—		13.9	21	14.1	23
A-WL	44.0	99	42.6	93	37.1	88

Heite等[22]研究了以相关系数作指标将峰位置按数值分类法分组，并以信息量作指标由每组选出熵值最大的“信道”，由此可得到相互较独立的信息量较大的红外峰位置，在此基础上再进一步重新编码。对5100个化合物组成的样本（ASTM红外光谱索引），在由140个光谱分段选出40个编码时，97.7%的编码是不重复的。Bink等[23]用主成分分析技术研究红外光谱的结构相关模式，用多元线性回归分析法分类，用可疑度作为分类时的优化指标。Ritter等[24]研究了互信息与最大似然分类器分类功能的关系。

第四节 定量分析的信息理论和方法

关于定量分析的信息理论基础，Eckschlager等曾经进行过较系统的研究，在“Collection Czechoslov. Chem. Communicationsz”杂志上发表过系列论文，并总结于其专著[25]中。在本节

中，将叙及这些研究的主要成果。

一、定量测定的信息量[25,26]

在进行定量测定之前，对被分析试样的成分的浓度范围往往并非完全一无所知。一般可假设待测组分 x 的含量在 $[x_1, x_2]$ 区间内，服从均匀分布，故其验前概率分布为

$$p_0 = 1/(x_2 - x_1)$$

在关于试样成分浓度范围确实一无所知的情况，则 $x_1 = 0\%$，$x_2 = 100\%$，上述假设仍成立。

在完成定量分析之后，分析结果一般服从正态分布，即 $x \sim N(\mu, \sigma^2)$。此处 μ 为试样中待测组分含量的真值，σ^2 为总体方差。则验后分布 $p(x)$ 是正态概率密度函数。用散度或卡尔贝克（Kullback）信息量的定义：

$$I(p /\!/ p_0) = \int_{x_1}^{x_2} p(x)\ln[p(x)/p_0(x)]\mathrm{d}x \tag{1-15}$$

为积分方便均取自然对数，故其单位为奈特（Nat）。对式（1-15）积分得

$$I(p /\!/ p_0) = \ln[(x_2 - x_1)/(\sigma\sqrt{2\pi e})] \tag{1-16}$$

实际上，在完成定量分析之后，分析结果一般用均值 x 表示，则此时可用学生分布来代替正态分布，当测定次数为 n，则分析结果的置信区间为

$$x \pm t_{\alpha,\varphi} \times (s/\sqrt{n})$$

在此，均分差 s（为 σ 的样本估计）可由下式求得

$$s = \sqrt{1/(n-1)\sum(x_i - x)^2}$$

$t_{\alpha,\varphi}$为学生分布的临界值，α 为置信率，$\varphi = n-1$ 为自由度。近似地，我们可认为分析测定后的后验分布为

$$p'(x) = 1/[2t_{\alpha,\varphi}(s/\sqrt{n})]$$

由此求得的信息量为

$$I(p' /\!/ p_0) = \ln[(x_2 - x_1)\sqrt{n}]/(2st_{\alpha,\varphi}) \tag{1-17}$$

从上式可以看出，由学生分布求出的信息量是一个测定次数的函数，一般说来，测定次数的增加且精密度高（测定次数增加时，s 不增大）可增加定量分析的信量。

如果分析结果存在系统误差 δ，即

$$\delta = |x - \mu|$$

在这种情况下，信息增益由下式给出

$$I(p_2 /\!/ p_0) = I(p /\!/ p_0) - I(p_2 /\!/ p)$$

在此，$p_0(x)$为待测分析物的验前分布，$p(x)$的意义同前，是服从 $N(\mu,\sigma^2)$的正态概率密度函数，而 $p_2(x)$则是服从 $N(\mu+\delta,\sigma^2)$的正态概率密度函数，在这里 $I(p_2/\!/p)$代表了以 p_2 代替 p 后引起的信息量的降低。经运算可得

$$I(p_2 /\!/ p_0) = \ln[(x_2 - x_1)/(\sigma\sqrt{2\pi e})] - [(\delta/\sigma)^2/2] \tag{1-18}$$

显然，当 $\delta = 0$ 时，仍得式（1-10）计算的 $I(p /\!/ p_0)$值。由此可见，分析方法的准确度影响定量分析的信息量。

二、提高分析精密度与准确度的信息量

前节已述及分析方法的精密度与准确度对信息量的影响。我们再进一步考察，当用一精

密度较差的分析方法（例如半定量分析方法）进行初步分析后，再用具有较高精密度与准确度的方法进行分析，将获得的信息量[28]。

设对未知样本作初步检验时，例如无机物分析，常用的这种半定量分析方法是发射光谱分析，初步分析所用的方法标准差为 σ_0，得到的分析结果均值为 μ_0，分析结果服从正态分布。如再以一精度较高的分析方法对试样作精密分析，如得到的结果的均值为 μ，标准差为 σ，显然 $\sigma\leqslant\sigma_0$，此时

$$p_0(x)=\int 1/(\sqrt{2\pi}\sigma_0)\exp\left\{-\frac{1}{2}[(x-\mu_0)/\sigma_0]^2\right\}$$

而

$$p(x)=\int 1/(\sqrt{2\pi}\sigma)\exp\left\{-\frac{1}{2}[(x-\mu)/\sigma]^2\right\}$$

散度或卡尔贝克(Kullback)信息量为

$$\begin{aligned}I(p\,/\!/\,p_0)&=\int_{-\infty}^{\infty}p(x)\ln[p(x)/p_0(x)]\mathrm{d}x\\&=\ln(\sigma_0/\sigma)+[(\mu-\mu_0)^2+\sigma^2-\sigma_0^2]/(2\sigma_0^2)\end{aligned}\tag{1-19}$$

式中，(σ_0/σ)反映了精密度提高的程度，兹记为 A，A 值恒小于 1，其值愈小。说明分析方法精密度提高愈多；$(\mu-\mu_0)^2/(\sigma_0^2)$ 则表征了提高精密度后对准确度的提高程度，记为 B。式（1-19）可改写为

$$I(p\,/\!/\,p_0)=\frac{1}{2}(A^2+B^2-1)-\ln(A)\tag{1-20}$$

式（1-20）表述了与精密度及准确度相关的因子 A 和 B 对信息量的影响。

三、痕量分析的信息量

在深入研究痕量定量分析时，发现分析组分的浓度极低时，定量分析数据以对数正态分布表述更为确切。本节将从这一点出发，探讨更确切地表述痕量定量分析信息量的方式[29,30]。

对数正态分布的分布函数为

$$\begin{aligned}F_\xi(x)&=p\{\xi\leqslant x\}=p\{\ln\xi\leqslant\ln x\}=\phi\{[\ln(x)-\mu]/\sigma\}\\&=1/(\sqrt{2\pi})\int_{\infty}^{[\ln(x)-\mu]/\sigma}\exp\{-[\ln(x)-\mu]^2/(2\sigma^2)\}\mathrm{d}x\end{aligned}$$

其相应概率密度函数为

$$\begin{aligned}f_\xi(x)&=F'_\xi(x)=1/(x\sigma)\varphi\{[\ln(x)-\mu]/\sigma\}\\&=1/(x\sigma\sqrt{2\pi})\exp\{-[\ln(x)-\mu]^2/(2\sigma^2)\}\end{aligned}\tag{1-21}$$

在考察痕量定量分析的信息量计算方法时，对待测组分的含量可以作出一大略估计，例如估计其含量最高值为 x_1，即 $x\in(0,x_1)$，即验前概率可表述为

$$p_0(x)\begin{cases}=1/(x_1-0)=1/x_1 & \text{当 } x\in(0,x_1)\text{时}\\=0 & \text{其他}\end{cases}$$

如用某一检测下限为 x_0 的痕量分析方法进行分析，结果待测组分含量在 x_0 以下，则验后分布为

$$p(x)\begin{cases}=1/(x_0-0)=1/x_0 & \text{当 } x\in(0,x_0)\text{时}\\=0 & \text{其他}\end{cases}$$

由此得出痕量分析结果待测组分未测出时的信息量为

$$I(p /\!/ p_0)_- = \ln(x_1/x_0) \quad (1\text{-}22)$$

由此可见，这时的信息量与分析方法的检测下限有关。x_0 愈低，信息量越大，

现考察痕量分析测出了待测组分含量时的信息量 $I(p /\!/ p_0)_+$。如前述，验后概率分布用对数正态分布表述较为确切。但由式(1-21)表示出的对数正态分布不能反映痕量分析测出待测组分时的一个事实，即此时 x 不能小于或等于 x_0。为解决此矛盾，用移位的对数正态分布，即令其概率密度函数为

$$p(x)\begin{cases}1/[(x-x_0)\sigma\sqrt{2\pi}]\exp\{-[\ln(x-x_0)-\mu]^2/(2\sigma^2)\} & x\in\langle x_0, x_1\rangle \\ =0 & \text{其他}\end{cases}$$

此处 μ、σ 是 $\ln(x-x_0)$ 的正态分布的均值和标准差。令

$$\mu = \ln(kx_0)$$

k 为移位对数正态分布的不对称系数。由此计算信息量得

$$I(p /\!/ p_0)_+ = \ln(x_1/x_0) + \ln[1/(k\sigma\sqrt{2\pi e})]$$

一般说来，有 $I(p /\!/ p_0)_+ \geqslant I(p /\!/ p_0)_-$，详细讨论可参阅文献[12]。

四、质量控制分析的信息量

在产品质量控制分析中，常需保证其所含某一组分的量保持在某一范围之内。设需控制的组分允许下限为 x_L，上限为 x_U，通过长期观测，可知 x 的波动的标准差为 σ_F，其与允许上下限的关系可用下式表述：

$$x_L = x - k\sigma_F \quad (1\text{-}23a)$$

$$x_U = x + k\sigma_F \quad (1\text{-}23b)$$

式中，x 为待控制组分的正常值；k 的取值决定了允许范围的宽度。

我们可设想多数产品的规格将落在允许范围 (x_L, x_U) 这一浓度区间，设落在这一区间的概率为 α，而其余落在此区间之外的概率则为 $(1-\alpha)$。故验前概率分布为

$$p_0(x)\begin{cases}=\alpha/(x_U-x_L) & \text{当 } x\in\langle x_L, x_U\rangle \text{ 时} \\ =(1-\alpha)/[100-(x_U-x_L)] & \text{当 } x\notin\langle x_L, x_U\rangle \text{ 时}\end{cases}$$

注意到上式中的常数取100，是因为 x 的取值范围只可能在 0～100%之间。经分析后的概率分布是正态的，即 $x\sim N(\mu,\sigma^2)$，且 $\sigma\leqslant\sigma_F$。

在进行分析后，如产品合格

$$I(p /\!/ p_0)_\alpha = \ln[(x_U-x_L)/(2\sigma\sqrt{2\pi e})]$$

如产品不合格

$$I(p /\!/ p_0)_{1-\alpha} = \ln\{[100-(x_U-x_L)]/[(1-\alpha)\sigma\sqrt{(2\pi e)}]\}$$

一般取 $0.90\leqslant\alpha\leqslant0.99$，所以 $\alpha\gg1-\alpha$，且 $(x_U-x_L)\ll100\%$，可以看出，$I(p /\!/ p_0)_\alpha$ 恒小于 $I(p /\!/ p_0)_{1-\alpha}$。即检查出不合格产品时的信息量较未检出时大。这是易于理解的。因在一般条件下，我们总是期待生产合格产品，否则生产无从正常运转。当发现了不合格产品时，较之证实生产在正常运转，分析工作的信息量较大。

前面的讨论中，产品合格范围的宽度是 $(x_U-x_L)=2k\sigma_F$，同时，如 $(x_U-x_L)\geqslant6\sigma$ 亦可满足时，即此时用于质量控制分析的分析方法的标准差必须小于 $1/3k\sigma_F$，那么，分析方法应至少能提供下述信息量

$$I(p /\!/ p_0)_{\min} = \ln[(2k\sqrt{n})/(\alpha\sqrt{2\pi e})] = \ln(\sigma/\sigma_F)\times(6\sqrt{n})/(\alpha\sqrt{2\pi e}) \quad (1\text{-}24)$$

式中，n 为产品质量控制分析的平行测定次数。

第五节 分析仪器的信息功能

在分析化学由于迅猛发展出现“理论危机”与动荡不安的 20 世纪 70 年代初，Kaiser[31,32]在一篇“元素分析的定量化”的著名论文中，首先提出了分析仪器与分析方法的供信能力的概念。Kaiser 的概念在分析信息理论发展中起了重要的作用，对于比较不同分析仪器与分析方法的功能具有实际意义，并可用作分析实验条件优化的指标。

一、分析仪器与分析方法的供信能力

分析仪器或分析方法的供信能力 $P_{\inf}$取决于可量测的量的数目 n，及每一量 i 可区分的阶梯数 S_i：

$$P_{\inf} = \sum_{i=1}^{n} \mathrm{ld}(S_i) \tag{1-25}$$

所谓可量测的量，可以是某一参量如频率、波长、质量数、电位等的不同取值。$P_{\inf}$的单位就是熵的单位，这实际上就是各种信号均认作等概率时的信号熵。对电磁波谱分析仪器，这种参量可以是波长或频率，例如发射光谱分析中谱线的波长代表一个参量，不同波长的谱线代表了可量测的量，如每一谱线的强度又划分为 10 个阶梯，则 S_i 为 10。

对于给定的参量 r，例如频率，不可能将其划分为无限个可分别测量的数值。限制这一划分的因素是所谓分辨率。对频率而言，分辨率在理论上是受到测不准原理制约的，对任意参量 r，可定义分辨率为

$$R(r)=r/\delta r$$

式中，δr 是最小的可互相区分的 r 值。

对于由参量的上限 r_U 到下限 r_L 的区间 Δr，将有 $\Delta r/\delta r$ 个不同的可量测的位置，即

$$\Delta r/\delta r=R(r)(\Delta r/r)$$

因此可定义

$$P_{\inf} = \int_{r_L}^{r_U} R(r)\mathrm{ld}[S_i(r)]\mathrm{d}r/r \tag{1-26}$$

如在 Δr 区间内 $R(r)$、$S_i(r)$可认作常数，即 $R(r)$、$S_i(r)$恒等于或它们的均值等于 R 和 S_i，则有

$$P_{\inf}=R\mathrm{ld}(S_i)\ln(r_U/r_L)$$

如再将时间作为独立变量考虑，例如时间分辨的分析仪器，设仪器的时间分辨能力为 $R(t)$，其含义类同于 $R(r)$，则上述定义可扩充为

$$P_{\inf} = \int_{r_L}^{r_U}\int_{t_L}^{t_U} R(r)R(t)\mathrm{ld}[S_i(r,t)](\mathrm{d}r/r)(\mathrm{d}t/t) \tag{1-27}$$

此处的 S 被认作是 r 和 t 二者的函数。如在积分区间 $R(r)$、$R(t)$及 $S_i(r,t)$均可认作常数，即取其均值表述，则有

$$P_{\inf}=R_rR_t\mathrm{ld}(S_i)\ln(r_U/r_L)\ln(t_U/t_L) \tag{1-28}$$

从式(1-28)可以看出，n 或分辨率 $R(r)$、$R(t)$对分析仪器或分析方法的供信能力的影响，远超过增加每一可量测量划分的阶梯 S 的影响。分辨能力可按数量级增加，达到例如 10^6 这样的数值，而当 S 由 2 增至 100 时，ld (S) 仅由 1 增至 7 左右。信息理论给我们提供了这样的

启示：为了提高分析仪器或分析方法提供信息的能力，如仅从提供的信息量数量这一角度考虑，增加可粗略量测的量（增加分辨率，增加第二个量测的参量）效果远超过改善量测的精密度。

表 1-7 列举了常见分析仪器及方法的供信能力[34]。表中的数值系按所涉及方法的一般实验条件估算，分子光谱的分辨能力的估算考虑了吸收谱带的带宽，除注明计数器读数外，均为模拟电路组成的仪器。

表 1-7　分析仪器与分析方法的供信能力

分析仪器或分析方法	R_t	r_U/r_T	R_τ	r_U/r_τ	S	$P_{\rm inf}$(或 $P_{\rm inf,t}$) bit
紫外可见分子吸收光谱	10	5			10^2	1.1×10^2
红外分子吸收光谱	100	10			10^2	1.6×10^3
紫外可见分子荧光光谱	10/10*	5/5*			10^2	1.8×10^3
紫外可见时间分辨分子荧光光谱	10/10*	5/5*	10^2	10^2	10^2	1.3×10^6
原子发射光谱，小型单色器	10^4	5			10^2	1.1×10^5
原子发射光谱，大型摄谱仪	10^5	5			10^2	1.1×10^6
法布理-彼罗特干涉仪	10^7	1.0002			10^2	1.4×10^4
原子荧光光谱，直读式，50 通道	50	--			10^2	3.5×10^2
原子发射光谱，时间分辨，小型单色器	10^4	5	10^2	10^2	10^2	7.7×10^7
原子荧光光谱，非分光（单通道）	1				10^2	7.0×10^0
原子荧光光谱，非分光，光子计数器（单通道）	1	—			10^4	1.4×10^1
X-射线荧光光谱，计数检测	10^4	5			10^4	2.2×10^5
火花源质谱	10^4	35			10^2	2.5×10^5
常规质谱	6	10			10^2	1.6×10^1
重量法		10			10^4	1.4×10^1

从表 1-7 可以看出，欲解决复杂的化合物成分与结构分析的问题，常需采用现代分析仪器手段。这一选择，是不以分析工作者的主观兴趣或偏爱为转移的，而是由现代分析仪器的强大供信能力所决定的。由于分析工作的仪器化、自动化、计算机化等而导致的分析化学学科的深刻变化，其根源亦在于现代生产与科研的发展，向分析化学提出的供给大量化学信息的日益增长的要求。

二、分析仪器的信道容量

本节拟讨论与分析仪器设计密切关系的信道容量问题。这一问题对分析仪器之间的联用，及分析仪器与计算机联机等具有重要意义。

根据 Nyquist 提出的采样定理，作为时间的函数 $G(t)$ 的分析信号，二次采样间隔时间 Δt_N 应为

$$\Delta t_N = 1/(2w)$$

式中，w 为分析信号的频率。

对于一个分析仪器，其产生或处理分析信息的速率由其信息量与采样速率决定。对于离散分析信号，如每秒输出 n 个符号，则“熵速率”可写成如下形式，单位为：bit · s^{-1}。

$$H = -n\sum p_i \mathrm{ld}(p_i)$$

对于连续分析信号，采样速率取决于输出信号的频宽，若其频率为 w (Hz)，根据前面讨论的 Nyquist 采样定理。最低必需的采样速率应为 $2w$(s)，否则将损失信息。因此，其“熵速率”为

$$H = -2w\int_{-\infty}^{\infty} p(y)\text{ld}[p(y)]\text{d}y$$

当输出的分析信号服从正态分布时，如其方差为 $V(=\sigma^2)$，则其“熵速率”为

$$H = 2w \times \text{ld}\ (\sqrt{2\pi e V}) \tag{1-29}$$

一个分析仪器的输出，可能是另一个分析仪器的输入，或是计算机的输入。因此，必须考虑这些认作信道的分析仪器或计算机传输与接受信息的能力，即所谓信道容量，或信道能传送信息的最大速率。在不考虑噪声项的情况下，式（1-29）实际就是分析信号平均功率即方差为 V 的正态分布信号时分析仪器的信道容量。

若分析信号与噪声（用 N 表示）均服从正态分布，合成的实际分析信号亦必服从正态分布，则由式（1-29）表示出的分析仪器的信道容量可改为

$$H(y) = 2w \times \text{ld}(\sqrt{2\pi e V + N})$$

同理，噪声的信道容量可表示为

$$H(y|x) = H_{噪声} = 2w \times \text{ld}(\sqrt{2\pi e N})$$

作为最大分析信息传输速率的分析仪器信道容量 C 可表示为

$$\begin{aligned} C &= \max[H(y) - H(y|x)] = w \times \text{ld}(1 + V/N) \\ &= w \times \text{ld}(1 + S/N) \end{aligned} \tag{1-30}$$

此处 S/N 为信噪比。

本节所讨论的问题的焦点，就在于研究几个分析仪器联用或分析仪器与计算机联机时从信道容量角度相互兼容的问题。这一问题与过滤一分析试液时，将其倾至一漏斗中情况颇类似。设滤纸折叠得当或漏斗的水柱效率高，可较快地将溶液滤过，否则速度减慢。若倾液速度超过滤纸和漏斗能接受与流过的容量，必然造成分析溶液的溢失。溢失分析试液即损失了化学信息，分析工作失败。在分析仪器之间或与计算机联机时，如信道容量不相匹配，会发生类似的损失分析信息的情况。接受前一个仪器送来的分析信息的仪器或计算机，其信道容量必须是不小于前一仪器的信道容量，否则将损失分析信息，整个分析系统是无法提供正确的结果的。

兹以红外光谱仪为例讨论式(1-30)的应用。

试比较红外光谱仪的两种类型：色散型（棱镜或光栅）及迈克耳逊-傅立叶（Michaelson-Fourier）型。一般的工作范围取 4000cm^{-1} 至 400cm^{-1}，典型的最小分辨波数为 8cm^{-1} 至 0.1cm^{-1}，红外光谱仪的分辨率与扫描观察时间一般可调，而信噪比 SNR 则是依赖于上述变量而变化的。色散型红外光谱仪有两个缺陷难于解决。一是由单色器分出单色光依次测定吸收，得到一张有一定精度的谱图所需时间较长，例如 2min。另一缺陷是由于使用达到所需分辨率必须用狭缝，因而进入单色器的光能不能太低，否则难于检测。采用迈克耳逊干涉装置的傅立叶变换红外光谱仪(FT-IR)可免此弊。由于色散型仪器中，信号的强度正比于观测时间（整个光谱观测时间 T 与分辨单元 M 之比值 T/M），而噪声则正比观测时间的平方根$[(T/M)^{1/2}]$，故信噪比 SNR 正比于$[(T/M)^{1/2}]$。而 FT-IR 光谱仪的信噪比正比于 $T^{1/2}$，因此在观测时间相同的情况下，FT-IR 光谱仪的信噪比为色散型仪器的 $M^{1/2}$倍，这一优点称费尔盖特(Fellgett)氏优点。获得这一优点所付出的代价，是对与之联机的计算机的信道容量显著增加。

现试用式（1-30）估算色散型红外光谱仪的信道容量。如设在吸收率接近 0%时信噪比 S/N 值有（500/1）左右，在 4000～400cm^{-1}光域，最小分辨单元设为 2cm^{-1}，为达到上述信噪比，设所使用的仪器需调整至慢的扫描速度，应用采样定理要考虑每一分辨单元需被采样 2 次，设由此计算采样速率，估算 w 值约 2.22 采样单元/秒，则信道容量 C_1 为

$$C_1 = 2.22 \times \mathrm{ld}(1+500/1) = 19.9$$

再看 FT-IR 光谱仪的情况，FT-IR 是从空间域采集数据．空间坐标（x）是迈克耳逊干涉仪中动镜的位移。动镜的运动可以 $x=vt$ 表述，此处 v 为动镜的速率，输出的信号称为干涉图。干涉图 b 是空间和时间的函数

$$b = b(x) = b(vt)$$

干涉图是许多频率的加合。每一频率与光谱波数 σ 之间存在一定关系：

$$b(x) = \int_0^{\infty} B(\sigma)\exp[-2\pi j\sigma x]dt \tag{1-31a}$$

或

$$b(\sigma t) = \int_0^{\infty} B(\sigma)\exp[-2\pi j\sigma vt]dt \tag{1-31b}$$

每一光谱波数 σ 对应一与动镜速度 v 相关的声频：

$$f(\mathrm{Hz}) = \sigma(\mathrm{cm}^{-1})v(\mathrm{cm/s})$$

采样频率由最高光谱频率 σ_U 决定，如 4000cm^{-1}，可得 $w=2500$Hz，如此时实际达到的 SNR 值为 6700/1，则信道容量 C_2 为

$$C_2 = 2500 \times \mathrm{ld}(1+6700/1) = 31776$$

FT-IR 光谱仪对与之联机的计算机的信道容量或运算速度要求显然要高得多。

有关作者[34]研究了色谱仪的信道容量，将式（1-18）用于色谱仪，w 值一般至多是数 Hz，但 SNR 可超过 10^{10}，故 C 可达到 50bit·s^{-1}左右，用色散型红外光谱仪与之联用，信道容量可能难于匹配，而用 FT-IR 光谱仪则完全可以满足要求。

三、分析方法的信息效率

对分析方法作出一个较全面的评价，需考虑其精密度、选择性、表面或空间的分辨能力、分析速度和成本等等诸多因素。精密度涉及标准差或相对标准差，对仪器分析而言，可以信噪比表述；选择性可以同时能分析的元素或化合物的数目表述，亦可以共存物质的影响表述；表面或空间分辨能力对现代材料分析甚为重要，因材料科学研究的新进展表明，许多材料的特异性能不但取决于某些成分的总含量，而且与这些成分在材料中的空间分布有关，分析速度以及分析成本等因素对解决实际分析课题很重要。除上面提到的各项指标外，一个根本性的指标是提供分析信息的功能，亦是本节讨论的中心课题，这里将引入一些补充概念。

有关作者[35]建议定义信息效率 E 为

$$E = \prod_{i=1}^{k} e_i \tag{1-32}$$

此处 e_i 为效率系数：

$$e_i = \begin{cases} \varepsilon_i & \text{如 } \varepsilon_i \leqslant 1 \\ 0 & \text{如 } \varepsilon_i > 1 \end{cases}$$

式中，ε_i 是效率比。对于分析方法的某一特性参数，效率比是指为解决某一分析问题（p）所需的值与该分析方法（A）所具有的数值之比。例如，如考虑的是信息量（I），为解决分析问题 p 需信息量 I_p，分析方法 A 能提供的信息量为 I_A，则 ε_i 值为

$$\varepsilon_I = I_p / I_A \tag{1-33}$$

不同情况下信息量的计算已在前面作过系统讨论。又如指标是指能测定的组分数，对给

定的分析问题需测定 N_p 个组分，分析方法实际能测定 N_A 个组分，则

$$\varepsilon_N = N_p / N_A \tag{1-34}$$

如指标是指分析时间，设分析工作需在时间 T_p 内完成，而该分析方法需时 T_A，则

$$\varepsilon_t = T_p / T_A \tag{1-35}$$

如对所有的指标均有 $e_i=1$，则 $E=1$，如某一指标小于 1，则 $E<1$；只要有一个指标达不到要求，$e_i>1$，即 $E=0$，该分析方法不能采用。

从另一方面，还可定义一与分析成本、信息量及信息效率相关的指标，权称信息效益：

$$R = M(E/\tau) \tag{1-36}$$

式中，τ 为分析成本，$M = \sum I_i$，为分析所有组分 i 的信息量之和。

按上述方法评估分析方法提供信息的效益，对给定分析方法而言，其值与分析问题的性质有关。对于解决某一问题信息效益高的方法，在解决另一课题时不一定有好的效益。

表 1-8 列举为解决测定钢中锰含量(0.05%～6%Mn)的方法比较，分析课题对于测定标准差的要求为 10%，分析时间的要求则设定三种情况：(a)短于 1d；(b)短于 30min；(c)短于 2min。由表 1-8 可见，对分析时间要求不高时，例如可允许在一天内完成分析工作(a)，容量分析、电位滴定与光度分析较光谱分析的信息效率与效益为高。这一点说明，现代分析化学的迅猛发展及随之而来的革命性变革，并不意味着分析工作者将抛弃原有的传统分析方法。这一显而易见的事实，颇类似于人类进入了航天飞行的时代，并不意味着应废止常规交通工具，或认为丧失步行能力对人类活动无关紧要。另一方面，表 1-8 亦显示，对要求快速完成的分析工作，在传统分析方法已不奏效时，仪器分析方法确有其优越处。

表 1-8 钢中锰含量的测定方法比较

分析方法	相对标准差/%	可同时测定组分 N_A	分析时间 T_A/min	相对成本（货币单位）	总信息量 (M)/bit	信息效率 E	信息效益 (P)/bit/单位成本
容量法（过硫酸盐法，二次 ZnO 分离）	2	1	60	16	4.00	(a) 0.018 (b) 0 (c) 0	0.0044 0 0
电位滴定	2	1	30	8	4.00	(a) 0.088 (b) 0.42 (c) 0	0.0044 0.21 0
分光光度分析法（转化为高锰酸根）	3	1	30	8	3.41	(a) 0.010 (b) 0.49 (c) 0	0.0044 0.21 0
原子吸收法	1	1	20	6	5.00	(a) 0.047 (b) 0.22 (c) 0	0.0039 0.19 0
发射光谱分析	5	50	3	1	133.8	(a) 0.00026 (b) 0.0013 (c) 0	0.0035 0.17 0
发射光谱-测微光度分析	10	50	30	12	83.8	(a) 0.042 (b) 0.020 (c) 0	0.0029 0.14 0
发射光谱-光量子计直读	5	50	1	20	133.8	(a) 0.0000087 (b) 0.00042 (c) 0.0063	0.000058 0.0028 0.042

参 考 文 献

1 Latinen H A. Anal Chem, 1980; 52: 606A

2 中国科学院长春应用化学研究所,"现代分析化学规划",1986

3 高鸿."分析化学现状与未来",《分析化学前沿》主题报告,北京,科学出版社,1991

4 国家自然科学基金委员会.自然科学学科发展战略调研报告·分析化学.北京:科学出版社,1991

5 Murray A. Anal Chem, 1991; 63: 271A

6 Korte.F. v. 1 Methodicum Chimicum, v. 1 Analytical Methods. Part A. New York: Academic Press, 1974. 2

7 Kateman G and Dijkstra A. Z Anal Chem, 1979; 247: 249

8 Kowalski B R. Trends in Analytical Chemistry, 1981; 1(3): 71

9 Mellanby K. "The Biological Significance of Analytical Data——All Enviromental Problems are Caused by Analytical Chemists!". Lecture Presented on 30th IUPAC Congress, Manchester: 1985

10 俞汝勤.化学计量学导论.长沙:湖南教育出版社,1991

11 高鸿."中国化学会成立五十周年大会报告",1980

12 俞汝勤.现代分析化学的信息理论基础.长沙:湖南大学出版社,1987

13 Eckschlager K and Stepanek V. Anal Chem, 1982; 54: 1115A

14 Eckschlager K and Stepanek V. Information Theory as Applied to Chemical Analysis, New York. John Wiley & Sons, 1977

15 Massart D L. J Chromatography, 1973; 79: 157

16 Grotch S L. Anal Chem, 1970; 42: 1214

17 Wangen L E, Woodward W S, Isenhour T L. Anal Chem, 1971; 43: 1605

18 Stenhagen E, Abrahamsson S, McLafferty F W(Ed.) Atlas of Mass Spectral Data. New York: John Wiley & Sons, 1969

19 Dupuis F, Dijksira A. Z Anal Chem, 1978; 290: 357

20 American Society for Testing and Materials, Codes and Instruction for Wyndotte ASTM Punched Cards, Philadelphia, Pa. 1964

21 Dupuis F, Dijksira A, Varder Maas J H. Z Anal Chem, 1978; 291: 27

22 Heite F H, Dupuis F, Klooster H A Van'T, Anal Chim Acta, 1978, 103, 313

23 Bink J C W G, Klooster H A Van'T. Anal Chim Acta, 1983; 150: 315

24 Ritter G L, Lowry S R, Woodruff H B et al. Anal Chem, 1976; 48: 1027

25 Eckschlager K, Stepanek V. Information Theory as Applied to Chemical Analysis, New York: John Wiley & Sons, 1997

26 Eckschlager K. Collec Czechoslov Chem Commun, 1976; 41: 2527

27 Eckschlager K. Collec Czechoslov Chem Commun, 1979; 44: 2373

28 Eckschlager K, Vajda I. Collec Czechoslov Chem Commun, 1974; 39: 3076

29 Eckschlager K. Collec Czechoslov Chem Commun, 1975; 40: 3627

30 Eckschlager K, Stepanek V. Microchimica Acta(Wien), 1978; 1: 107

31 Kaiser H. Anal Chem 1970; 42(2): 24A

32 Kaiser H. Anal Chem 1970; 42(4): 26A

33 Winefordner J D. in "Trace Analysis, Spectroscopic Methods for Elements", Winefordner J D Ed., New York: John Wiley & Sons, 1976; 435~442

34 Huner J F K and Smit H C. Z Anal Chem, 1969; 245: 84

35 Danzer K and Eckschlager K. Talanta, 1978; 25: 275

第二章　化学计量学中常用的名词与术语

第一节　分析信息与分析采样

一、分析信息理论与方法

ASTM 红外光谱索引（ASTM infrared spectral index）　对红外光谱数据库进行编码，所有文件包括了约 102000 条红外光谱，按 Wy-andotte-ASTM 法编码，将 (2.0～15.9)μm 波长区域划分为宽度为 0.1μm 的 140 波长段，在每一波长段，如吸收强度超出选定的规定临界值，则编码为 1，否则为 0。由 ASTM 文件编码了 96900 条红外光谱（WL-1）。

编码（coding）　编码是通讯理论中常用的一种方法，在化学计量学中常借用来对波谱进行编码。如在质谱中，设每一质量数位置仅区别有峰（其编码为 1）与无峰（其编码为 0），这种编码方式常称为 0-1 编码。此时，每个信道的最大可提供的信息量为一个比特（ld2，在此 ld 表示以 2 为底的对数）

编码信息量（coding information）　从信息理论来考虑，不同的编码方式所含信息量不同。对于质谱的 0-1 编码，每个信道 i 的信息量即熵值。N 张质谱图中出现 1 或 0 的概率均为 0.5 时，该信道的信息量为 1 比特。当质谱的 n 个信道都相互不相关时，此时对 N 个质谱的 0-1 编码的总信息量为 n 比特。

编码效率（coding efficiency，CE）　通过编码实际可得信息量与最大可得编码信息量之比称为编码效率。当被编码的每个信道都相互不相关时，编码的总信息量为最大可得编码信息量。但实际编码由于各信道之间存在相关性，实际可得信息量将恒小于最大可得编码信息量。研究如何进行编码能提供最大的信息量是编码理论的一个重要内容。

比特（bit）　对于一个只有两种结果出现的随机事件，如果它们出现的概率均为 0.5，那么，该事件的信息量就是一个比特（$\log_2 2$）。

不确定度（uncertainty）　不确定度是随机事件的特性，信息理论中熵的概念就是一种随机事件不确定度的度量。

部分因子设计（fractional factorial design）　针对 2 水平全因子设计提出了部分因子设计思路，一般都具有 2^{n-d}的形式，其中 2 表示因子的水平数，n 为因子数（或称因素的数目），d 为可能缩小试验次数的 2 的乘方数，当 d 为 1 时，因试验次数可以缩小一半，故有半因子设计法之称；所谓半因子设计，实际上就是 2^{n-1}次因子设计。

分析信息理论（analytical information theory）　分析化学是一门化学信息科学。在化学的各个分支学科中，分析化学担负的任务与其他分支学科的不同之处，就在于分析化学的研究对象，不是直接提供某种具体的例如无机和有机材料，而是提供与这些材料的化学成分和结构相关的信息，研究获取这些信息的最优方法与策略。分析信息理论是一种从信息理论出发，用信息理论来看待分析过程所获信息量大小和用信息理论来评价分析仪器所供信息量多寡的一种方法。

供信能力（informing power）　分析仪器或分析方法的供信能力 P_{inf}实际上就是分析仪器或分析方法所能提供的信号熵。它取决于可量测的量的数目 n，及每一量 i 可区分的阶梯数 S_i：

$$P_{inf} = \sum_{i=1}^{n} ld(S_i)$$

所谓可量测的量，可以是某一参量，如频率、波长、质量数、电位等的不同取值。P_{inf}的单位就是熵的单位，这实际上就是各种信号均认作等概率时的信号熵。对电磁波谱分析仪器，这种参量可以是波长或频率，例如发射光谱分析中谱线的波长代表一个参量，不同波长的谱线代表了可量测的量，如每一谱线的强度又划分为 10 个阶梯，则 S_i 为 10。

互信息（mutual information） 互信息是信息论中一个很重要的概念，它的定义式为

$$I(A;B)=H(A)-H(A|B)=H(B)-H(B|A)=I(B;A)$$

式中，$I(A;B)$称为 B 关于 A 的互信息，或$I(B;A)$称为 A 关于 B 的互信息。式中 H 表示熵。同时，上式也可称为信息守恒定律或信息平衡原理。

可疑度（equivocation） 可疑度是分析信息理论中的一个概念，其定义为：在 B 试验发生后 A 事件的熵，即 $H(A|B)$，实际表述了在进行了试验 B 以后的试验 A 的“不确定度”，$H(A|B)$就称为可疑度。这样定义的可疑度，实际上可以说是反映了分析仪器或方法提供信息的能力与解决给定分析课题的需要之间的差距的定量度量。

熵（entropy） 熵是信息论和热力学中一个十分重要的概念，它是事物不确定度和混乱度的一种度量，其数学定义如下：

$$H = -\sum_{i=1}^{k} p_i \log p_i$$

在信息理论中，仙农给出了上述定义并称 H 为**熵**。log 一般表示以 e 为底，此时所得熵的单位为奈特（nat）；如果以 10 为底，其单位为的特（dit）；而以 2 为底时，其单位则为比特（bit）。对于确定性事件，其熵为零；对于等概率结果的事件，其熵最大。

散度（divergence） **散度**由 Kullback 提出，故又称为卡尔贝克（Kullback）信息量。当一个分布（p）被另一个分布（q）取代时，其定义为

$$I(q \,//\, p) = \sum q_i \log(q_i / p_i)$$

其中 p 为概率验前分布、q 为验后概率分布。对于连续变量的散度或卡尔贝克（Kullback）信息量的定义为

$$I(q \,//\, p) = \int q(x) \log(q(x)/p(x)) dx$$

条件熵（conditional entropy） 条件熵是信息理论中的一个重要概念。其定义为：如果试验 A 出现结果 A_k 时，试验 B 在此条件下的熵或不确定度为条件熵。更进一步，还可定义在进行试验 A 的前提下试验 B 发生条件下的条件熵。如果随机试验 A 与 B 相互独立，则它们同时发生的熵等于它们条件熵之和。

分析仪器的信道容量（information channel capacity of an analytical instrument） 一个分析仪器的输出，可能是另一个分析仪器的输入，或是计算机的输入。因此，我们必须考虑这些认作信道的分析仪器或计算机传输与接受信息的能力，即所谓信道容量或信道能传送信息的最大速率。在不考虑噪声项的情况下，实际就是分析信号平均功率即方差为 V 的正态分布信号的分析仪器的信道容量。

信息效率（information profitability） Danzer和 Eckschlager 建议定义信息效率 E 为

$$E = \prod_{i=1}^{k} e_i$$

此处 e_i 为效率系数：

$$e_i=\begin{cases}\varepsilon_i & 如\ \varepsilon_i\leqslant 1\\ 0 & 如\ \varepsilon_i>1\end{cases}$$

在此，ε_i 是效率比。对于分析方法的某一特性参数，效率比是指为解决某一分析问题（p）所需的值与该分析方法（A）所具有的数值之比。例如，如考虑的是信息量（I），为解决分析问题 p 需信息量 I_p，分析方法 A 能提供的信息量为 I_A，则 ε_i 值为

$$\varepsilon_i=I_p/I_A$$

二、分析采样理论与方法

采样理论（sampling theory） 采样理论是指如何进行试样采集的数学统计理论。怎样通过局部采样可在统计意义上尽可能的代表总体，是采样理论和方法所研究的内容。分析化学中常用的采样方法包括固体物质的采样方法、动态过程的采样方法和质量检验的采样方法。

采样常数（sampling constants） 为表征实验室样本的均匀性，或考虑样本的分隔效应，Ingamell 和 Visman 分别定义了几个采样常数，如均匀度常数和反映分隔（segregation）程度的常数，以更好地描述采样特征。

代表性采样（representative sampling） 代表性采样一般指特定的分析项目所涉及的采样。例如按环境保护部门规定采集废水试样就是这种例子。在分析化学的实际工作中，代表性采样是一种分层采样的特殊情况，这种情况的分层采样可对目标成分提供总体均值的无偏估计。

分层采样（stratified sampling） 当分析对象可划分为若干采样单元时，随机采样可以是总体的全体采样，亦可以是分层或分步采样。分层采样是先将分析对象划分成不同的部分或层，然后对不同的层次进行随机采样。

计量抽样检验（sampling inspection by variable） 计量抽样检验是一种质量检验的方法。这种检验方法一般是针对定量测试产品中某一变量的分布来进行的。

计量抽检特性曲线（operating characteristic curve in sampling inspection by variable） 分析检验中常见的情况是测定某一组分的百分含量，例如某种有效成分的最低含量，某种有害成分的最高含量等。在考虑这种检验的采样时，首先需确定合格产品的相应定量标准。今假设待测的组分含量这一变量服从正态分布，并设产品中的有效成分 x 的最低含量标准为 x_0，分析方法的标准差为 σ，可计算对应于任意真实含量的 x 的 u 值，即标准正态分布值：

$$u=(x_0-x)/\sigma$$

从标准正态分布可计算实际含量为 x 时该批试样被接受的概率 p。将概率 p 对 x 作图，可得计量抽检特性曲线，简称 OC 曲线。

计数抽样检验（sampling inspection by attributes） 计数抽样检验也是一种质量检验的方法，多用于以件计数的产品检验，如采用化学分析方法，则是作定性分析提供检验对象是否合格的属性。例如用看谱镜确定某种杂质存在或不存在。对于一批共 N 个试样，如其中不合格的占 $p\%$，合格的占 $q\%$，取 n 个试样找到 c 个不合格试样的概率 P_c 可按超几何分布计算。

颗粒性质因子（particle property factor） Gy对非均匀体系的采样发展了一套较全面的理论。根据 Gy 的采样理论，固体采样的方差公式引入了颗粒形态因子（f）。如果是在理想的情况下，颗粒形态因子 f 可以取 1；如果所有颗粒都是以 μ 为直径的小圆球组成时，则颗粒形态因子 f 可取 0.524。如果总体物质是具有特殊形态的颗粒，如黄金颗粒，颗粒形态因子 f 可以低至 0.2。

系统采样（systematic sampling） 系统采样系指为了检验某些系统假设而采集的试样，例如生产或其他过程中成分随时间、温度的变化而在空间中变化，这种场合下的采样问题有重要的实际意义。一般是间隔一定区间（时间、空间、区域）采样。间隔不一定是等距的，有时事先可预期总体成分是不均匀的，系统采样要尽量减少这种不均匀性的影响．对于这样的情况，可采用分层采样。系统采样的误差分析与随机采样是相似的。

最小采样数目（minimum number of samples） 在采样理论中，在给定采样方差的条件下的最小采样量称为最小采样数目，一般以质量来计。

第二节 化学试验设计与优化

半因子设计法（half-fraction factorial design） 针对 2 水平全因子设计提出了部分因子设计思路，一般都具有 2^{n-d}的形式，其中 2 表示因子的水平数，n 为因子数（或称因素的数目），d 为可能缩小试验次数的 2 的乘方数．当 d 为 1 时，因试验次数可以缩小一半，故有半因子设计法之称；所谓半因子设计，实际上就是 2^{n-1}次因子设计。

饱和设计（saturated designs） 在实验设计中，实验次数（实验点数）等于所要确定的未知参数的个数的实验设计，称为饱和设计。

饱和 *D*-最优设计（saturated *D*-optimal design） 在*D*-最优设计方法中，如能使得做到实验次数（实验点数）等于所要确定的未知参数的个数的 *D*-最优设计，称为饱和 *D*-最优设计。

变尺度法（variant scale method） 变尺度法可看成是局部最优方法中梯度法和牛顿法的一种拓广。梯度法收敛太慢；牛顿法要计算二阶导数矩阵和矩阵求逆，工作量大。变尺度法的基本思路就是利用迭代逐步逼近二阶导数矩阵的逆矩阵。因这类方法的搜索方向是不断变化的，故有变尺度法之称。变尺度法有多种，目前用得较多的是 DFP 法（由 Davidon-Fletcher-Powell 提出）和 BFGS 法（由 Broyden-Fletcher-Goldfarb-Shanno 提出）。

残差正态图（residual normal plot） 因为正态分布为一般表示由多种不定因素而产生出来一种分布，是随机噪声的特征，所以，回归所得残差应服从正态分布。残差正态分布图就是一种很好的检验方法，如所得残差可用正态分布图中的一条直线表出，则说明模型是合理的。

产生子（generator） 在因子试验设计中，一个构成半因子设计的简便方法就是先设定一个同名和别名效应，然后就用这个别名效应来产生别的各个别名效应，以构成一个半因子设计，这样设定的别名效应就可称之为产生子。

***D*-最优设计**（*D*-optimal design） *D*-最优设计由统计学家 Kiefer 提出，是一种基于使试验设计的信息矩阵的行列式值达到最大，从而使试验设计的所得模型的参数估计很准确的一种试验设计的方法。它对于所研究体系的模型已知的情况最为适合。

单纯形优化（simplex optimization） 在s 维空间 R^s 中，单纯形是指具有 $s+1$ 个顶点的多面体，若各个棱长彼此相等，则称为正规单纯形。单纯形优化法是以单纯形顶点为其起始研究对象，通过比较，淘汰其中最差的点，继利用反射，构成新的单纯形，继续优化搜索，以达到寻优目的。单纯形优化方法是一种局部最优方法。化学家们对原始的简单单纯形法进行了许多改进，在化学研究中应用较广。

单纯形试验设计法（simplex experimental design） 单纯形试验设计法是一种序贯试验设计方法。它以单纯形优化方法为基础进行试验设计或寻优，是化学计量学中试验优化方法之一，它通过构建单纯形来安排试验，通过比较这些试验结果，淘汰其中指标值最差的试验点，在可能改进试验指标的方向新增一个试验点，再作一次试验，继续进行单纯形搜索。此

种设计方法具有灵活和容易实现，又可序贯实行的优点，在化学研究中应用很广。

单纯形格子点（simplex lattice point） 在s维试验空间中，s因子的混料区域$\mathfrak{R}$可用$(s-1)$维单纯形来表示。单纯形上的点，如其s个坐标中有1个坐标为1，其余$(s-1)$个坐标为1，则这样的点称为单纯形顶点。当$s=3$时，其单纯形是一个高为1的等边三角形，它的三个顶点的全体称为一阶格子点集，记为$\{3, 1\}$。如将等边三角形的三边各二等分，则此三角形的三个顶点与三条边的中点的全体称为二阶格子点集，记为$\{3, 2\}$，其中共有6个点。如将等边三角形各边d等分，类似地可得d阶格子点集，记为$\{3, d\}$。如此定义的单纯形格子点集，又简称为单纯形格子点。

单纯形格子设计（simplex lattice design） 根据单纯形格子点，Scheffe提出了单纯形格子设计方法，这种设计方法可以保证设计点分布均匀，且回归系数的计算变得简单，是一种有效的混料设计方法。设有s分量系统，各分量$x_i(i=1,2,\cdots,s)$的变化范围满足约束条件$\sum x_i=1$，当采用d阶典型的Scheffe多项式回归模型时，试验点选为$\{s, d\}$的C_{s+d-1}^{d}个格子点，这正好与所采用的d阶典型的Scheffe多项式回归方程中待估计的回归系数的个数相等，故单纯形格子设计是饱和设计，是在“试验次数最少”意义下的最优设计。

单纯形重心设计（simplex gravity center design） 单纯形重心设计也是一种混料试验设计。单纯形重心设计方案的试验点数等于相应的s阶重心多项式回归方程中待估系数的个数，故单纯形重心设计也是饱和设计。试验点的选取方法是取单纯形顶点及它们的一些重心点。对$s=3$，其单纯形重心设计共有7个点；对$s=4$，其单纯形重心设计共有15个点。对s因子混料设计系统，前面所说的单纯形顶点及顶点的各种重心点总数为

$$C_s^1+C_s^2+C_s^3+\cdots+C_s^s=2^s-1$$

当采用s阶重心多项式回归模型时，应取这全部2^s-1个点作为试验点。显然，一、二阶单纯形格子设计与一、二阶单纯形重心设计相同，当$d>2$时，对于同阶多项式回归模型来说，单纯形重心设计的试验点数要少于单纯形格子设计的试验点数，且d越大其差越大。

等价定理（equivalence law） 等价定理是最优设计中的一个很重要的定理，它说明了D-最优实验设计与G-最优试验设计等价，而且G-最优试验设计的最优值就等于未知参数的个数，为构建D-最优实验设计提供了思路。

定义关系（defining relation） 在因子设计中，可事先定义某个效应A与某项交叉效应$ZMPD$同名，$A=ZMPD$，这样，我们可得到一个定义关系或定义相反（defining contrast），即$I=ZMPD\times A=ZMPDA$。利用这个定义关系，可产生所有其他别的各个同名效应来。

多步法（multistep method） 多步法是数值求解常微分方程的一类方法。数值求解常微分方程方法的基本思路是将时间变量离散化，即产生一离散点序列t_0，t_1，t_2，t_3，…，以及变量区间$h_n=t_{n+1}-t_n$。然后，在每一点t_n处，用先前求得的值计算，并用该计算值作为该时间点解的近似值。这样的方法在微分方程数值解方法中称为差分法（也称逐步法或离散变量法）。差分法提供了一个利用k个先前得到的数值y_n，y_{n-1}，y_{n-2}，…，y_{n-k+1}来计算y_{n+1}的法则，称为k步法。如果$k=1$，则是单步法，如果$k>1$，则是多步法。

二次饱和D-最优设计（second-order saturated D-optimal design） 对于二次回归模型

$$y=\beta_0+\sum_{j=1}^{s}\beta_j x_j+\sum_{i\leqslant j}^{s}\beta_{ij}x_i x_j$$

试验点个数与未知待估参数个数$m[m=(s+1)(s+2)/2]$相等的D-最优设计称为饱和D-最优设计。有人证明，对$s\geqslant 7$不存在饱和D-最优设计。当$s=2$、3时，饱和D-最优设计存在，

但关于 $s \geqslant 4$ 的饱和 D-最优设计，至今尚未解决。

混杂 (confounding) 在部分因子试验设计中，主效应与交叉效应，或交叉效应与别的交叉效应会出现具有完全一致的符号的现象。这种两种效应具有同一符号的情况，在部分因子设计中，统计学家将其称为混杂或共现现象，这些同符号的效应将称为同名或别名效应。

共轭方向 (conjugate direction) 共轭方向为局部优化算法中的一种重要寻优方向。设 $\boldsymbol{A}$ 为一个 $(n \times n)$ 对称正定矩阵，$\boldsymbol{p}$、$\boldsymbol{q}$ 为两个 n 维向量，若 $\boldsymbol{p}^t\boldsymbol{A}\boldsymbol{q}=0$，则称 $\boldsymbol{p}$、$\boldsymbol{q}$ 为两个共轭向量，如果 $\boldsymbol{A}$ 为一个单位矩阵，则 $\boldsymbol{p}$、$\boldsymbol{q}$ 两向量正交。对于正定二次 n 元函数，如果依次沿 n 个相互共轭的方向进行一维搜索，则最后得到的就是它的最优点。如果一个算法的搜索方向是互相共轭的，则称为共轭方向法，以后讨论的共轭梯度法，DFP 法都是共轭方向法。

好格子点法 (good lattice point) 好格子点 (glp) 法是一种基于数论的产生网格点的方法。它首先从 glp 产生表中取出产生向量 $(n; h_1, \cdots, h_s)$，n 为点集中的样本点数，s 为所考察空间的维数。继而用下式产生 glp 集 $\{x_k,\ k=1,\ \cdots,\ n\}$，
$x_k=[\{(h_1k-0.5)/n\},\{(h_2k-0.5)/n\},\cdots,\{(h_sk-0.5)/n\}],k=1,\cdots,n$。这里 $\{.\}$ 代表了取 $\{\}$ 中的小数部分。由好格子点法产生的格子点具有很好的均匀性。

混料试验设计 (mixture experimental design) 混料试验设计是一种存在有约束条件的试验设计方法。在化工、材料、冶金、食品、精细化工等领域的产品大多是几种化学成分混合后加工制造出来的，它们的某项或某几项特性指标与混合成分所占比例有关。如何确定各成分在混料（或称配方）所占比例，以使我们感兴趣的特性指标在一定条件下达到最优，是生产和科研中的一个重要问题。混料试验设计是专门针对这一方面研究的试验设计方法。在混料问题中，试验的响应值仅与每种组分的百分比有关，而与混料的总量无关。

基于数论方法的序贯优化法 (sequential number-theoretic optimizarion method) 基于数论方法的序贯优化法是一种将数论方法 (NTM)，或称为伪蒙特卡洛法 (quasi Monte-Carlo methods)，和数值分析相结合的一种寻优方法。它采用 NTM 在搜索空间均匀布点搜索最优点，继序贯收缩搜索空间而不断搜索的一种全局优化算法。与其他全局优化算法比，SNTO 具有原理清晰明了，算法易于实现等特点。

交互效应 (interactive effect) 交互效应是试验设计中一个十分重要的概念。在试验设计的建模过程中，有必要进行析因分析，如所研究的对象为一线性模型，则主效应的估计就说明了因素本身对于模型影响的大小，可是，在绝大多数的实际问题中模型很难用线性模型表达，都存在着的交互效应，即因素与因素之间的交互影响，所以，在估价了它们自身的主效应之后，一般都需要继续估价交互效应。

交互效应表 (interaction effect tables) 交互效应表是正交试验设计中用来标明正交表中哪些列与哪些交叉项具有混杂或共现的一种表格。通过它人们可容易地有效地安排正交试验。

局部优化算法 (local optimal method) 局部优化算法是一类经典的优化算法，由于它们一般只能得到局部最优解，故有局部优化算法之称。局部优化算法中如按需不需要对目标函数求导来分，可分成两类，即一类需要对目标函数 $f(x)$进行求导运算，这样的方法包括梯度法、牛顿法、共轭梯度法以及变尺度法，这类方法充分利用了函数的解析性质，它们的收敛速度多数比不用求导的优化方法（直接法）要快；而另一类则毋需求导，常称直接法，它们是在化学计量学中得到广泛应用的单纯形法、坐标轮换法和 Powell 方法。

均匀性准则 (criterion for uniformity) 度量均匀性的准则很多，其中偏差是使用历史

最久，为公众所广泛接受的准则。设 $x_1, x_2, \cdots, x_n$ 为 [0,1] 的 n 个点，任一向量 $x=(x_1, x_2, \cdots, x_n) \in [0,1]$，记 $v(x)=x_1 \times x_2 \times \cdots \times x_n$ 为矩形 $[0,x]$ 的体积，n_x 为 $x_1, x_2, \cdots, x_n$ 中落入矩形 $[0, x]$ 的点数，则

$$D(x_1, x_2, \cdots, x_n)=\max_{x \in [0,1]} |n_x/n - v(x)|$$

称为点集 $\{x_1, x_2, \cdots, x_n\}$ 在 [0, 1] 中的偏差。

均匀设计 (uniform experimental design) 均匀设计是一种着重在试验范围内考虑试验点均匀散布以获得最多试验信息的较新的试验设计方法。它特别适合于具有较多水平的试验和对模型没有先验信息的非线性系统。

均匀设计表 (uniform design table) 均匀设计与正交设计相似，也是通过一套精心设计的表来进行实验设计的。类似于正交试验设计，均匀设计表一般用 $U_n(q^s)$ 来表示，其中 n 为试验次数，q 为水平数，s 为因子数。

龙格-库塔法 (Range-Kutta method) 龙格-库塔法是一种最常用的微分方程的数值解法。它只需知道一阶导数而毋需明确定义或计算其他阶导数。近似值由若干次函数求值得到。当步长 $h \to 0$ 时，经典的四阶龙格-库塔法通过阶为 h^4 的各项渐近地与台劳级数法相吻合。龙格-库塔法编程较易，且对很大一类数值计算问题是十分稳定的。

盲目搜索 (blind search) 盲目搜索是人工智能中的一种搜索方法。在人工智能中，对于很简单的问题，直接的方法就是进行“盲目搜索”，这是很费时的办法，让计算机依次产生并检验各结点可能出现的分支，随着搜索过程的进行，搜索树不断长大。搜索树的每个结点可赋予一个水平，根结点位于 0 水平，接着是水平 1，余类推。水平的层次代表了搜索的深度。广度优先搜索和深度优先搜索是两种应用较广的盲目搜索方法。

模拟退火法 (simulated annealing method) 模拟退火法是一种随机的全局优化算法。其主要思路是将退火思想引入优化领域，它源于对固体退火过程的模拟，采用 Metropolis 接受准则，并用一组称为冷却进度表的参数控制算法进程，使算法在多项式时间里给出一个近似全局最优解。固体退火过程的物理和统计性质是模拟退火算法的基础背景，Metropolis 接受准则使算法具有跳离局部最优的机制，而冷却进度表的合理选择是算法成功的重要前提。

拟水平均匀设计 (pseudo-level design) 在一般的均匀设计表中，各因子的水平数都是相等的，由于在实际的化学实验和化工生产中，有时各个因素的水平数是不同的，需要使用混合水平设计表。拟水平均匀设计就是一种通过引入所谓拟水平技术，直接用均匀设计表来构造拟水平均匀设计表的一种针对混合水平试验的均匀设计的方法。

牛顿法 (Newton method) 牛顿法是一种局部最优的、最古老、最基本的算法。其主要思路是对梯度法的搜索方向进行改进，即加进目标函数的二阶导数矩阵的信息，以加速收敛速度。对于二次型目标函数，用牛顿法只须迭代一次就可得到最优点。对于非二次函数，由于它们在极小点附近和二次函数很接近，使用牛顿法，其收敛速度也是很快的。但是牛顿法要求初始点选得比较好，如离最优点较远时，就容易发生迭代发散或收敛到非最优点的情况。

偏差 (discrepancy) 偏差是一种用于比较点集在试验空间的均匀性的度量。度量均匀性的准则很多，其中偏差是使用历史最久，为公众所广泛接受的准则，其定义如下：设 x_1，x_2，$\cdots x_n$ 为 C^m 的 n 个点，任一向量 $x=(x_1, x_2, \cdots, x_n) \in [0,1]^m$ 的，记 $v(x)=x_1 \times x_2 \times \cdots \times x_n$ 为矩形 $[0,x]$ 的体积，n_x 为 $x_1, x_2 \cdots, x_n$ 中落入矩形 $[0,x]$ 的点数，则

$$D(x_1, x_2, \cdots, x_n)=\sup |n_x/n - v(x)|$$

$$x \in [0,1]^m$$

称为点集 $\{x_1, x_2, \cdots, x_n\}$ 在 $[0, 1]^m$ 中的偏差。

Powell 法 (Powell method) Powell法是一种局部最优算法，又称方向加速法，其基本思路是利用逐次构造共轭方向，并以此来加速收敛，故有方向加速法之称。Powell 法构造共轭方向毋需计算目标函数的导数，这是该法的一个显著性优点。

全局优化算法 (global optimization methods) 全局优化算法是相对局部优化算法的一类寻优方法。与局部优化算法的关键不同之点就在于它们都具有跳出局部优化点的机制，在化学计量学研究中，它们的最主要的代表性算法有模拟退火算法、遗传算法和基于数论方法的序贯优化法。

四分之一因子设计法 (quarter fraction factorial design) 四分之一因子设计法是一种部分因子设计法，通常以 2^{n-2}的形式表示出，因其与全因子设计的试验次数相比，只有全因子设计试验次数的四分之一，故有四分之一部分因子设计法之称。

梯度法 (gradient method) 梯度法（或称最速下降法）是求解优化问题的一种最古老、最基本的局部最优算法。其主要思路是取函数的负梯度方向（最速下降方向）作为迭代的搜索方向。

同名或别名 (alias) 在部分因子设计中，一些主因子和一些交叉效应将存在重复的机会，这种两种效应具有同一符号的情况，在部分因子设计中，统计学家将其称为共现或混杂现象，这些同符号的效应将称为同名或别名效应。

线性图 (linear graph) 对于正交设计的试验安排，除交互效应表外，还有一个很重要的工具，就是正交设计表的线性图。对于不同的正交表，存在其相应的标准线性图。图中的点及其相应的数码代表正交表的列，图中的连线及其相应的数码也代表正交表中的某一列，只是这样的列是表示其线条相连的两个点所代表的两列的交互效应。线性图的顶点所对应的列，可用于安排试验的各个因素。

一次饱和 *D*-最优设计 (first-order saturated *D*-optimal design) 一次饱和 *D*-最优设计是饱和 *D*-最优设计的一种方法。对于 *s*-维立方体 $-1 \leqslant x_j \leqslant 1$，($j=1,2,\cdots,s$) 上的一次回归模型，

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+\cdots+\beta_sx_s$$

选取 ($s+1$) 个点组成实验设计方案就称为一次饱和 *D*-最优设计。

遗传算法 (genetic algorithm) 遗传算法最早由 Holland 于 20 世纪 60 年代创建，被用于解决化学上的难题。遗传算法是一族用计算机模拟生物自然进化过程随机搜索最优解的方法。它引入达尔文进化论，通过模拟自然界生物“遗传→变异→适者生存”的进化过程，对优化空间进行随机搜索，从而得到全局最优解。数值遗传算法是将待优化的各个参数排列在一起，当作一条染色体，每个参数即为染色体中的遗传基因。根据染色体对环境的适应性（即其目标函数值的一个转换指标），通过各种遗传操作控制其繁殖情况，淘汰差的，保留好的，最终得到最好的染色体，亦即全局最优点。遗传算法的过程一般可概括为五个基本步骤：① 按设定的规模产生第一代群体，群体中每条染色体的长度由所需优化的参数的个数决定；② 评价染色体的适应性，通过一定的方式将每条染色体中的参数值代入所求目标函数中，求出对应函数值；③ 将目标函数值经过转换，得出染色体的适应性优劣；④ 有选择地交叉染色体，每次随机地从母代染色体中选出两条染色体，随机地交换它们某些对应位置上的数段，从而产生下一代的新个体（在选母代染色体的过程中，每条染色体被选中的概率与其适应性成正比）；⑤ 变异，按一定概率随机地对染色体上的基因进行变异，由此可以产生一些新的个体。

整个算法在第②步和第⑤步之间循环。

因子设计分辨率（resolution of factorial design） 因子设计的分辨率是因子试验设计中的一个很重要的概念，一般说来，分辨率越高的部分因子设计越好。因子设计的分辨率的正式定义为：设计的分辨率为 R 的部分因子设计为 p，在这样的因子设计中，没有 p 项交叉效应与（$R-p$）项交叉效应存在共现现象。值得指出的是，这一原始定义由 Box 给出，但现在有人提出，此定义还并不完善，需要修改。部分因子设计在文献中的表述一般为数字或罗马字，例如对于 2^{5-1} 部分因子设计的分辨率可表示如下：

$$2^{5-1}(\mathrm{V}) \quad ; \quad 2^{5-1}(\mathrm{IV})$$

上述符号就分别表示分辨率为 5 和分辨率为 4 的 2^{5-1} 部分因子设计。

对于上述给出的分辨率，一般都遵循以下三条原则：

① 一个具有分辨率为 3（$R=\mathrm{III}$）的部分因子设计，其主效应之间不存在共现现象，但是主效应一般都与二项交叉效应存在共现关系。

② 一个具有分辨率为 4（$R=\mathrm{IV}$）的部分因于设计，其主效应之间不存在共现现象，同时主效应与二项交叉效应也不存在共现现象，但是主效应一般都与三项交叉效应存在共现关系。

③ 一个具有分辨率为 5（$R=\mathrm{V}$）的部分因于设计，其主效应之间，主效应与二项交叉效应及三项交叉效应之间不存在共现现象或混杂关系，但是主效应与四项交叉效应，二项交叉效应与三项交叉效应之间却存在着混杂关系。

正交设计（orthogonal design） 正交试验设计从本质上来说，可以算成是一种部分因子设计方法，因其主要特征是基于正交表来进行试验设计和数据分析，故有正交试验设计之称。

正交设计表（orthogonal arrays） 正交设计表是正交试验设计用来安排实验的一种特殊表格。一个矩阵的任意两列都搭配均衡，则称矩阵 $\boldsymbol{A}$ 为一个正交表。这里称矩阵为表，因其可写成表格的形式，常简记为

$$\mathrm{L}_n(t_1 \times t_2 \times t_3 \times \cdots \times t_m)$$

L 是正交表的代号，来源于拉丁方这一试验设计表的第一个字母、n 表示共作 n 次试验，而 $t_j(j=1,2,\cdots,m)$ 则代表第 j 列由 t_j 个水平组成。如所有 t_j 均相等，则可记为 $\mathrm{L}_n(t^m)$，称为 t 水平正交表。如两列水平数不相等时，则称为混合型正交表。用正交表来安排试验，可以达到以很小的试验次数来最大限度的获取有关各因素主效应及交叉效应的信息。

主效应（main effect） 在试验设计中，每一种实验因素对于实验结果的贡献，就称为试验效果的主效应。它们可通过方差分析得到，也可通过回归分析得到。

最优试验设计（optimal design） 最优试验设计是一类试验设计的方法。它们以一个最优指标为其试验设计的目标来构建试验设计点，其中 D-最优设计是统计学中应用最广泛的一种最优试验设计方法。

坐标轮换法（coordinate changing method） 坐标轮换法是一种局部最优算法。对于 n 维实数空间的目标函数 $f(\boldsymbol{x})$，其算法的基本思路就是依次沿其 n 个坐标轴方向进行一维寻优，经多次循环迭代后以达到最优解。

第三节　多元校正与多元分辨

不等性方差（heteroscedastic variance） 不等性方差噪声是指量测噪声随分析信号大小而变化，分析信号大的量测噪声也大，例如在紫外可见光谱和质谱的量测中，量测噪声的绝对值在一定范围中随量测信号而增大。这种噪声就是不等性方差噪声，其方差在不同的量测

点上的方差是不同的。在多元校正和多元分辨中要加以注意。

崩溃点（breakdown point） 为评价统计方法的稳健性，Hodges 引入了崩溃点的概念。一般说来，崩溃点的概念可解释为强烈影响估计偏离其"实际"情况的最小的奇异点数与估计时所包含点数的比值。最小二乘估计的崩溃点是 $1/n$，这就是说，只要在 n 个量测点中有一个是奇异点，它就可以完全破坏最小二乘估计。如果存在坏杠杆点，最小一乘的崩溃点也是 $1/n$!

病态体系（I 11-conditioned system） 在直接多元校正的模型中隐含着一个假设，即不同的化学物质的量测谱是不同的，即校正矩阵 $\boldsymbol{X}$ 必须是一个满秩矩阵。可是，在分析化学的实际中往往存在着这样的情况，在某些光谱（如最常用的紫外可见谱）的量测中，有些不同的化学物质（如某些异构体）却具有很相似的光谱，虽然它们的光谱多少有些不同，但却非常相似，这就会使校正矩阵 $\boldsymbol{X}$ 接近亏秩，而使满秩条件很难保证，或者说纯物质谱之间的差别已与量测误差相近，这样的情况在数学上称之为病态。这样的分析体系称为病态体系。

标准加入迭代目标转接因子分析法（standard addition iterative target transformation factor analysis） 经典分析化学中经常采用标准加入法，以克服基体或底液对分析校正的影响。事实上，标准加入的方法同样可推广到多元校正的情况。多元线性回归和 Kalman 滤波算法都是首先得获得校正所需的敏感度矩阵或是各物种的标准谱，如果事先无法获得标准谱，即由于待测样本的基体条件不清楚，无法找到标准谱的量测条件时，即可采用通用标准加入法来进行多元校正。通用标准加入法的实质是在待测多组分体系中直接加入多种待测标准物，以构成一个新的量测矩阵，再从这新的量测矩阵求得在此基体溶液中的各标准物的标准量测谱，继用求得的标准谱对待测多组分体系进行直接校正，即可求得此多组分体系的各组分的浓度。

残差双线性分解法（residual bilinearization） 残差双线性分解法是一种多元校正方法。其基本思路仍是利用双线性纯组分矩阵秩为 1 的基本原理，先是把含未知干扰的样本矩阵按式 $\boldsymbol{Y}=c_1\boldsymbol{X}_1+c_2\boldsymbol{X}_2+\cdots+c_n\boldsymbol{X}_n+\boldsymbol{R}+\boldsymbol{E}$ 所示那样分成两部分，一部分为待测的分析物（$\boldsymbol{X}_i$，$I=1, 2, \cdots, n$)，另一部分为未知干扰背景 $\boldsymbol{R}$，继交替对数据进行两步分析，首先在固定背景的基础上对分析物进行校正；然后固定得到校正的分析物相对浓度，对干扰背景进行主成分分解。经不断交替进行上述两个过程，直至背景干扰的分解可在已知主成分数的基础上完全分解，且校正的分析物相对浓度达到收敛。

乘幂法（power method） 乘幂法为一种快速简便的求取特征值和特征向量的数值算法，实质上它与化学计量学中常用的非线性迭代偏最小二乘算法，亦称 MIPALS 算法是同一的。它通过迭代方法来逼近特征向量的同时又得到特征值。一般说来，对于特征值差别较大的特征向量，收敛速度很快，迭代 10～20 次一般即可收敛，但当接近量测噪声所对应的特征向量时，因它们的特征值相差较小，收敛速度将显著减慢，有时甚至很难收敛，故在算法中常采用固定迭代次数的办法，如通常定为 30 次，来终止迭代。

纯分析信号（net analytical signal，NAS） 纯分析信号由 Lorber 提出，用来定义化学计量学中多元校正的各类分析化学中常用的品质因数。其物理意义为分析物量测矢量在共存干扰矢量正交补空间的模，可看成是扣除共存干扰矢量后的分析物的纯分析信号。

纯分析物种的秩（net analyte rank，NAR） 在张量校正中表示待测物种的纯分析物种的秩，定义为

$$NAR_N=\mathrm{rank}(\boldsymbol{M})\text{-}\mathrm{rank}(\boldsymbol{M}|\boldsymbol{N}),$$

次轴（secondary axes） 次轴是由 Malinowski 提出的因子分析误差理论中一个重要概

念，表示属于纯粹由量测误差而引出的那部分特征向量所组成的因子轴，一般说来它们所对应的特征值都很小，Malinowski 将它们称为次轴。

抽出误差（extracted error，XE） 在因子分析中，量测误差被分为两项，即所谓的置入误差（IE）和抽出误差（XE）。其中抽出误差（XE）表示属于纯粹由量测误差而引出的那部分特征向量所组成的因子轴，即次轴之中，属于可用因子分析方法剔除的那部分误差。同时，它与置入误差（IE）和量测误差即真实误差（RE）的关系，可由一勾股定理的方式表示，即

$$(RE)^2=(IE)^2+(XE)^2$$

固定尺寸移动窗口渐进因子分析法（fix-sized moving window evolving factor analysis） 固定尺寸移动窗口渐进因子分析法是用于求联用色谱产生的二维数据局部化学秩的一种局部因子分析方法。其思路相当简单，反渐进因子分析法逐步扩大主成分分析窗口的做法，它却固定一个窗口的尺寸，只是不断将此窗口从二维数据的起点移向终点，从而得到一个特征值的对数值对保留时间的作图，以得到二维色谱数据在保留时间方向的信息。

窗口因子分析法（window factor analysis，WFA） 窗口因子分析法是一种多元分辨方法，其主要思路是利用一个只有（$n-1$）个物种的“窗口”的信息，从而求出相对应的第 n 个物种的纯色谱，以达到分辨的目的。

等性方差（homoscedastic variance） 等性方差系指噪声的标准偏差在数据中是处处相等的，一般线性回归模型都有这种假设。

得分矩阵（score matrix） 对数据阵进行奇异值分解后，一般可得到三个矩阵，其中那个列正交的矩阵在化学计量学中常称为得分矩阵。一般说来，得分矩阵反映了样本与样本之间的关系。

迭代目标转换因子分析法（iterative target transformation factor analysis，ITTFA） 迭代目标转换因子分析法是一种多元分辨方法，其主要思路是利用主成分分解的抽象矩阵来进行迭代，以求得一个逼近的可能解，从而达到分辨的目的。

多元校正（multivariate calibration） 多元校正是通过利用化学量测谱的多变量信息，来对未知混合物体系进行浓度校正的一种化学计量学方法。对于白色分析体系，它可分为直接多元校正、间接多元校正、广义标准加入法和广义内标校正法。

多元分辨（multivariate resolution） 多元分辨是针对二维或更高维分析数据进行未知物分析的化学计量学方法。例如，通过利用可产生二维数据的一类仪器产生的数据，包括联用色谱或者二维荧光仪，可对未知混合物体系进行组分数的测定，还可进一步对复杂的未知多组分体系进行定性定量分析。这种方法主要用来解析黑色分析体系。

多元线性回归方法（multiple linear regression） 多元线性回归方法为基于最小二乘估计的统计解析方法，可对线性模型的未知参数进行估计，是一种直接校正方法，也可用于定量结构与活性关系的研究。

非线性迭代偏最小二乘算法（nonlinear iterative partial least squares，NIPALS） 非线性迭代偏最小二乘算法是乘幂法的一种变种，是化学计量学中常用的一种用来求特征值和特征向量的有效算法。

杠杆点（leverage point） 杠杆点在统计学中又称为强影响点。它是通过俗称为帽子矩阵的 $\boldsymbol{H}$ 矩阵（$\boldsymbol{H}=\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t$）的对角元素 h_{ii} 来定义的。一般说来，当 h_{ii} 大于它的平均值的 2 倍或 3 倍时，可认为第 i 个量测值所对应的 $\boldsymbol{x}_i$ 是一个强影响点或称为杠杆点。杠杆点具有很强的相互掩蔽效应，使得在进行回归诊断时，由于杠杆点的存在，可造成很多假象而掩蔽真

正的奇异点和杠杆点。

广义秩消失因子分析法（generalized rank annihilation factor analysis） 广义秩消失因子分析法是一种多元校正方法。其主要思路是将秩消失因子的计算问题转变成为广义特征方程求解的问题，从而使得秩消失因子分析法不再局限于一次只能测一个待测分析物的约束，此外，该法还可拓广至其他类型的情况。

惯量因子（momentum factor） 在人工神经网络中最主要就是确定连接权重，Rumelhart 等人提出了多层前传网络误差反传算法(BP 算法)。它根据误差反馈过来的情况以调节网络的连接权重，从而达到优化目的。一般都是用 $\Delta w_{ij}=\rho\delta_i u_j$ 来调节权重，在此，ρ 为步长因子，可在（0～1）之间取值；u 为第 j 节点上的抽象变量，一般为输出层或是隐蔽层的节点，δ 为梯度因子。步长因子 ρ 的取值大小对收敛速度有很大的影响，如果取值太大，则可能引起迭代过程的振荡；反之，若取值太小则会导致权重调节的迭代过程收敛太慢。一般说来，为了加快迭代过程且防止迭代过程的振荡，在 $\Delta w_{ij}=\rho\delta_i u_j$ 中引入一个惯量因子 λ，而使得

$$\Delta w_{ij}(\alpha+1)=\rho\delta_i u_j+\lambda\Delta w_{ij}(\alpha)$$

以保证迭代收敛速度。

荷载矩阵（loading matrix） 任一实数矩阵经奇异值分解后，可得三个矩阵，即

$$\boldsymbol{X}^{n\times d}=\boldsymbol{USV}^t$$

其中，矩阵 $\boldsymbol{V}^t$ 为一行正交矩阵，在化学计量学中则被称为荷载矩阵。

活性函数（activation function） 在人工神经网络的计算中，输入层与隐蔽层或隐蔽层与输出层都需通过一个活性函数 f(.)把它们联系起来。活性函数一般可以取以下几种形式：

① 符号函数 $f(u)=\mathrm{sgn}(u)=f_b(u)$

当 u 为正数时，取 1，否则取－1，此函数用于处理模式识别问题时很有用。

② s 型非线性函数 $f(u)=(1+e^{-u})^{-1}=f_s(u)$

其输出值介于 0～1 之间，进行非线性校正或预测时很有用。

③ 非线性反正弦函数 $f(u)=\tanh(u)$

其输出值在＋1 和－1 之间，对模式识别和非线性回归或预测都可用。

加权最小二乘回归法（weighted least squares regression） 加权最小二乘回归法是一种专门针对在量测中存在不等性方差的多元校正方法。因为最小二乘回归和 Kalman 滤波法一般都假设量测误差是服从零均等方差的正态分布，然而，在分析化学的量测中，有时这一假设并不可能满足，即量测值在不同的量测范围内，量测精度往往不同，因而量测误差也不尽相同。这种情况在光谱分析中很常见。加权最小二乘回归法的基本思路就是对不同的误差项 e_i^2 加不同的权，即将目标函数改为 $f(c)=\sum w_i e_i^2$，这样，当量测值的精度高时，w_i 大，反之，w_i 小。这样可以使拟合的曲线非常接近于精度高的点，保证拟合的准确度高。

交叉校验法（cross validation） 交叉校验法是一种很有效的模型校验方法，在统计学中又称为刀切法。其主要思路是通过多次剔除一个或多个训练集的样本建模，继以由此而得的模型来预测被剔除的一个或多个样本，遍历所有被剔除的样本后得到它们的总预测误差，就以此作为模型校验和选择的依据。在化学计量学中一般用来确定主成分分析和偏最小二乘的主因子数。

经典诊断方法（classic diagnostic methods） 诊断方法主要是研究那些对最小二乘估计有较大影响的奇异点的特性，旨在找到一种作图或计算方法来直接先将这些奇异点或强影响点检出，以消除它们对回归参数估计的影响。而经典诊断方法主要是基于一些非稳健估计，如

最小二乘估计模型和它的残差、均值及协方差矩阵的分析，故在统计学中有经典诊断方法之称。常用的经典诊断方法有用帽子矩阵 $\boldsymbol{H}$ 诊断强影响点或杠杆点，用马氏距离和 Cook 距离诊断奇异点等。

间接校正方法（indirect calibration methods） 间接校正方法是一类多元校正方法的总称。其基本特征是用一些已知浓度的混合物经某种实验设计方法首先构成一校正矩阵 $\boldsymbol{Y}$，然后，利用这校正矩阵找出这些混合物量测谱所对应的各组分纯物种的量测谱（K-矩阵法）或是利用它找到混合物量测谱与各混合物浓度的一种线性关系（P-矩阵法，主成分回归，偏最小二乘），这样一类方法因其不直接采用纯物种的标准量测谱来进行校正，故统称为间接校正方法。因间接校正方法是用混合物量测谱来直接估计未知混合物量测谱的各组分浓度，所以，有一定减轻非线性因素的功效。但因间接校正方法仍是采用线性模型，所以对于严重非线性分析体系仍不可能得到令人满意的结果。

局部因子分析（local factor analysis） 局部因子分析是一种强调对数据矩阵的局部结构进行分析的因子分析方法。它与全局因子分析的不同之点就在于它不是对整个数据矩阵来进行因子分析，而是只对部分感兴趣区域进行因子分析，如对选择性区域进行因子分析就是一个例子。

渐进因子分析法（evolving factor analysis） 渐进因子分析法是一种多元分辨方法，它巧妙地利用了二维数据的一个重要化学特点，即化学物质在很多情况中的依次出现特性，具有这样特性的二维数据包括联用色谱（如 GC-IR，HPLC-DAD，GC-MS 等）数据、滴定光谱数据（如在不同 pH 值的光谱）、动力学光谱数据等。渐进因子分析法的思路很简明，它利用主成分分析所得的特征值为其基本出发点，不同之处只是逐步渐进地延拓至整个矩阵，从而得到各物种在该数据矩阵中的出现点和消失点。

矩阵校正方法（matrix calibration methods） 在化学计量学的多元校正和多元分辨中，求得的解是否唯一依赖于从分析仪器所得的数据维数，一般说来，分析仪器所得的数据维数越高，则得到的信息将越多，所以在多元校正和多元分辨中，有基于矢量（一阶张量）、矩阵（二阶张量）和三阶张量的不同算法。在多元校正中凡属基于矩阵类型的数据的算法都统称为矩阵校正方法。如对于灰色体系，只有矩阵校正方法可得到唯一解。

矩阵分辨方法（matrix resolution methods） 由于在化学计量学的多元校正和多元分辨中，可求得的解是否唯一依赖于从分析仪器所得的数据维数，一般说来，分析仪器所得的数据维数越高，则得到的信息将越多，所以在多元校正和多元分辨中，有基于矢量（一阶张量）、矩阵（二阶张量）和三阶张量的不同算法。在多元分辨中凡属基于矩阵类型的数据的算法统称为矩阵分辨方法。

K-矩阵法（K-matrix method） K-矩阵法是一种间接校正方法，其主要思路是先通过混合物的校正矩阵借最小二乘法求得各组分的纯物种谱，然后再利用如此求得的纯物种谱去求得未知待测混合物的各组分的浓度。这样做有两个好处：① 因各组分的纯物种谱是由混合物的校正矩阵求得，故有一定的抗非线性功效；② 因校正矩阵可自行确定，故可很方便进行实验设计。K-矩阵法的缺点是它必须进行两次求逆，这将使误差传递扩大。

Kalman 滤波（Kalman filtering） Kalman滤波为信号处理中一种最优线性递推滤波方法，因属递推算法，具有计算速度快且所需计算机内存少的特点，故在工程实践中，特别在航空空间技术中得到广泛应用。Kalman 滤波引入化学计量学后，在分析化学计量学的多元校正中得到广泛的应用。Kalman 滤波算法在化学中可用于很多方面，如化学动力学的速率常数

的测定，非平衡状态的电位滴定，多组分化学体系的直接校正等。在多元校正中，Kalman 滤波法是一种直接校正方法。

岭回归法（ridge regression） 与基于最小二乘的多元线性回归不同，岭回归是一种有偏统计回归算法，其基本公式为：

$$\hat{c}(k)=(X'X+kI)^{-1}X'y$$

式中，X 为直接校正矩阵，y 为混合物的量测矢量，$\hat{c}(k)$为混合物的浓度估计矢量，I 为$(n\times n)$阶单位矩阵，k 为可调参数。由于 k 的引入，岭回归失去了最小二乘回归所具有的无偏估计的特点，故称为有偏估计。

岭迹图（ridge trace plot） 岭迹图是根据式 $\hat{c}(k)=(X'X+kI)^{-1}X'y$，用 $\hat{c}(k)$对 k 直接作图即可得到。岭回归法中一般就采用岭迹图来决定 k 值。

***M*-估计法**（*M*-estimation） M 估计是一类极大似然估计的简称。其主要思路也是改变估计的最小二乘回归的目标函数。反最小二乘估计用残差平方和作为目标函数的做法，它重新定义了一个目标函数：

$$\mathrm{Min}\sum\rho(r_i)$$

式中的 $\rho(.)$为一偶函数，即 $\rho(-t)=\rho(t)$，且在 t 等于零时具有唯一的最小。如定义 $\Psi(.)$为$\rho(.)$的导函数的话，实际上，M 估计是直接从构造 Ψ 函数而不是构造 ρ 函数出发，不同的 M 估计有着不同的 Ψ 函数。M 估计在实际的计算中是一个加权最小二乘的迭代过程，即

$$\alpha^{(k-1)}=[X^{t}(W^{(k-1)})^{2}X]^{-1}X^{t}(W^{(k-1)})^{2}y$$

其中对角矩阵 $W^{(k+1)}$的元素由 Ψ 函数所确定。

帽子矩阵（hat matrix） 帽子矩阵是统计学回归诊断中一个很重要的概念。在 $y=X\alpha$ 的线性回归模型中，帽子矩阵可定义为：$H=X(X'X)^{-1}X^{t}$，这个矩阵之所以称为帽子矩阵是因为它可直接将量测矢量 y 转换成为最小二乘的估计值，即

$$\hat{y}=Hy=X(X'X)^{-1}X^{t}y=X\alpha$$

帽子矩阵可用于诊断回归分析中的强影响或称为杠杆点。

***P*-矩阵法**（*P*-matrix method） P-矩阵法是以 $C=PY$ 为其数学模型，直接以校正矩阵的浓度阵为预测目标，求得回归系数矩阵 P 从混合物量测矩阵就可直接求出混合物的组分浓度，这样就避免了两次求逆过程，简化了 K-矩阵法。

偏最小二乘法（partial least squares） 主成分回归方法的主要思想是直接对混合物量测矩阵 Y 进行正交分解，继而用浓度矩阵 C 对被分解的正交主成分进行回归。偏最小二乘法则更进一步，它不但对混合物量测矩阵 Y 进行正交分解，而且在分解 Y 的同时将浓度矩阵C 也进行正交分解，这样浓度矩阵的主成分也被分离出来：

$$Y=USV^{t}=U^{*}S^{*}V^{t*}+E_{Y}=T^{*}V^{*t}+E_{Y}$$

$$C=PGQ^{t}=P^{*}G^{*}Q^{t*}+E_{C}=R^{*}Q^{t*}-E_{C}$$

这里，由混合物量测矩阵 Y 分解得到的矩阵 T^{*} 以及由浓度矩阵 C 分解得到的矩阵 R^{*}，代表了除去大部分噪声后的响应和浓度的信息。而且，在进行同时分解时，考虑了矩阵 T^{*} 和矩阵 R^{*} 之间应有的线性关系，这就构成了偏最小二乘法不同于主成分回归的主要之点，即在分解矩阵 Y 时考虑矩阵 C 的因素，而在分解矩阵 C 时考虑矩阵 Y 的因素，交互效验相互影响，通过迭代时交换迭代矢量而使两个分解过程合二为一。偏最小二乘法是化学计量学中应用得最广泛的一种方法。

全局因子分析法（glohal factor analysis） 全局因子分析法是简单利用整个量测矩阵来

进行因子分析的方法。它基本不利用量测矩阵的局部性质。

前传网络误差反传算法（back-propagation，BP 算法） 前传网络误差反传算法是人工神经网络在化学计量学中应用较广泛的一种方法。这一算法在模式识别及非线性回归方面得到了广泛的应用。所谓误差反传的基本思路就是定义一个误差函数作为训练网络的目标函数，然后采用一种方法根据误差反馈过来的情况以调节网络的连接权重，从而达到优化目的。

人工神经网络（artificial neural networks） 人工神经网络的研究从 20 世纪 50 年代末、60 年代初开始，代表性工作是 Rosenblattd 的感知器（perceptron）和 Widrow 的自适应线性元，至今已有 30 余年历史。特别是 80 年代以来，以 Hopfield 等人的联想记忆网络模型、制约优化网络模型、Rumelhart 等人的多层前传网络误差反传算法（BP 算法）等为代表的一批优秀工作的出现，推动了人工神经网络研究。人工神经网络具有很强的非线性映照能力，是发展较快的化学计量学研究领域。人工神经网络的基本思路是基于人脑细胞（神经元）的工作原理来模拟人类思维方式，以建立模型来进行分类与预测的。人工神经网络方法模拟人脑的神经元、通过神经纤维或突触与别的神经元相联系，用以接受来自别的神经元的信息，并将信息继续传给别的神经元的网络系统，借用神经元来表示一个计算单元，它也可以像神经元一样通过网络与别的计算单元相连，构成网络系统，以达到非线性映照的能力。人工神经网络的方法既可用于校正，又可用于化学模式识别。

s 型非线性函数（sigmoid nonlinearity function） 在人工神经网络的计算中，其输入层与隐蔽层或隐蔽层与输出层都需通过一个活性函数 $f(.)$把它们联系起来。s 型非线性函数是化学计量学中用得最多的一种活性函数，它的表达式为：$f(u)=(1+e^{-u})^{-1}=f_s(u)$，其输出值介于 0～1 之间，进行非线性校正时很有用。

投影旋转因子分析法（projection rotation factor analysis，PRFA） 投影旋转因子分析法是针对相分辨荧光光谱仪（PRFS）所产生的三维数据的解析面提出的，是一种基于张量数据的多元分辨方法。其主要思路仍是基于因子分析法，不同的是它巧妙地利用了张量数据的性质来进行投影旋转，从而可获得有物理意义的唯一解。

通用标准加入法（generalized standard addition method） 经典分析化学中经常采用标准加入法，以克服基体或底液对分析校正的影响。事实上，标准加人的方法同样可推广到多元校正的情况。通用标准加人法的实质是在待测多组分体系中直接加入多种待测标准物，以构成一个新的量测矩阵，再从这新的量测矩阵求得在此基体溶液中的各标准物的标准量测谱，继而用求得的标准谱对待测多组分体系进行直接校正，即可求得此多组分体系的各组分的浓度。

稳健主成分回归（robust principal component regression） 稳健主成分回归是通过对主成分回归进行稳健化而得。其中一个例子是由 Walczak 和 Massart 提出的一个稳健主成分回归法（RPCR），它旨在构造一种奇异点的稳健检出方法。其基本思路是用椭球多变量修剪法（MVT）来得到一个稳健的离差矩阵，继而在此无奇异点干扰的离差矩阵上进行主成分分析以检出 $\boldsymbol{X}$ 矩阵的奇异点，构成此稳健主成分回归法的第一步。接着，采用最小中位方差估计（LMS）法所得的标准化残差来检出量测矢量 $\boldsymbol{y}$ 中的奇异值。

稳健偏最小二乘法（robust partial least squares） 偏最小二乘估计（PLS）为化学计量学中一种最常用方法。将其稳健化则成了稳健偏最小二乘法。如将 PLS 方法中的对 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 矩阵的荷载矢量 $\boldsymbol{w}$ 和 $\boldsymbol{c}$ 的估计过程稳健化，结果表明，对于 $\boldsymbol{Y}$ 矩阵中的奇异点此算法是有效的。这稳健化的代价是有关荷载矢量 $\boldsymbol{w}$ 之间正交关系难以保持。此算法是为对 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 矩阵的奇异点进行独立地消除而设计的，其中的稳健步骤是采用了由 Beaton 和 Tukey 提出的双加权法。

稳健诊断方法（robust diagnostic methods） 因为经典的诊断方法大都是基于一些不稳健的统计量，如均值、协方差等之上，故对多个奇异点和有强影响点存在时的诊断效果不好，于是，统计学家就提出一些新的稳健诊断方法，这些稳健诊断方法的特点就在于它们大都是基于一些稳健的统计量之上，故诊断效果均优于经典的诊断方法。

稳健距离（robust distance） 为了克服奇异点的掩蔽效应，Rousseeuw 等提出了一个稳健距离，

$$RD_i=(x_i-T(x))C(X)^{-1}(x_i-T(x))^t$$

将此式与马氏距离 $MD_i=(x_i-\bar{x})C^{-1}(x_i-\bar{x})^t$ 进行比较，容易发现，此式与其的主要差别在于用 $T(x)$ 和 $C(X)$ 分别代替了其中的不稳健的均值矢量 $\boldsymbol{x}$ 和协方差阵 $\boldsymbol{C}$，故有稳健距离之称。

稳健主成分分析（robust principal component analysis，RPCA） 稳健主成分分析是表示一类将主成分分析稳健化的方法。由于将主成分分析稳健化可用不同的方法，在化学计量学中形成了多种稳健主成分分析方法。

稳健奇异值分解法（robust singular value decomposition，RSVD） 基于投影寻踪，Ammann 提出了一种稳健奇异值分解法。此方法由两步迭代过程所构成：① 对数据矩阵进行最小二乘回归拟合，继而将其向回归超平面进行旋转；② 通过稳健的广义 M 估计得到协方差矩阵的特征值和特征矢量，从而找到描述该数据阵的稳健协方差阵和期望值。此稳健奇异值分解法可以成为很多多元分析方法的基础，如稳健多元回归、稳健多元判别分析和主成分分析等。

因子分析（factor analysis） 因子分析是化学计量学中一个应用得十分广泛的多元统计方法，其主要思路是通过对数据矩阵进行正交分解而提取出主因子或特征变量来替代原始的数据变量，从而找到隐含在数据中的重要信息的一种解析方法。它与主成分分析十分类似。

因子分析的误差理论（error theory of factor analysis） 因子分析的误差理论的主要思路是将分解所得的因子分成所谓主轴和次轴，主轴包含前 d 个大特征值所对应的特征向量；而次轴属于纯粹由量测误差而引出的那部分特征向量，一般说来它们所对应的特征值都很小。由此而将数据矩阵中的系统信息和量测噪声分开。

因子设计（factorial design） 因子设计由著名统计学家 Box 等提出，是一种多因素的全面试验设计方法，它通常可由多种因素的水平数的一个乘积表来表示。如一个三因子二水平的因子设计可由 2^3 来表示，亦即 2×2×2，需进行 8 次试验；而一个四因子三水平的因子设计则可由 3^4 来表示，亦即 3×3×3×3，需共进行 81 次试验。因子设计的任务就是要通过这样的试验安排来了解各个因素及各因素水平之间的搭配对响应值或指标的影响。

隐蔽层（hidden layer） 隐蔽层为人工神经网络的一个很重要的概念。对于一个具有多于三层的人工神经网络，它表示存在于输出层和输入层的网络层。因它们不是人工神经网络的外层，故有隐蔽层之称。

有偏估计方法（biased estimating method） 有偏估计方法是一类很有用的估计方法，有时又称压缩估计。由于共线性的影响，使得最小二乘估计很不稳定，被估参数具有很大方差。为减小被估参数的方差，统计学家就提出了一系列压缩估计的方法，可减小被估参数的方差，但是与此同时它们却失去了最小二乘估计的无偏性，故有有值估计方法之称。最常见的有偏估计方法包括岭回归、主成分回归和偏最小二乘回归法。

约束背景双线性分解（constrained background bilinearization） 约束背景双线性分解法的基本思路是利用样本量测矩阵 $\boldsymbol{Y}$、校正矩阵 $\boldsymbol{X}_i$（$i=1$，…，n）和未知干扰背景矩阵的秩

这些已知条件，将灰色分析体系的解析问题转化成为一个优化问题，而求得有物理意义的唯一解的一种基于矩阵数据的灰色分析体系的多元校正方法。

子窗口因子分析法（subwindow factor analysis） 子窗口因子分析法是一种基于矩阵数据的多元分辨方法。与渐近因子分析法、窗口因子分析法及正交投影分辨法相同都是以分辨纯组分色谱为其目标，不同的是子窗口因子分析法是以分辨纯组分光谱为其目标。其主要思路是利用在色谱方向上不同子窗口含有一个且只含有一个相同纯物质光谱的特性，从而将该纯物质光谱分辨出来。

主轴（primary axes） 主轴是因子分析误差理论中一个重要概念，表示属于由系统信息而引出的那部分特征向量所组成的因子轴，一般说来它们所对应的特征值都显著大于误差水平，Malinowski 将它们称为主轴。

张量分辨方法（tensor resolution methods） 对于黑色分析体系的解析，要获得唯一解，需要三维，即张量类型的数据。张量分辨方法就是这样一类可对张量类型的数据进行分辨解析的化学计量学方法。广义秩消失因子分析法、投影旋转因子分析法、三线性交替分解法，满秩平行因子分析法等都属于张量分辨方法。

张量校正理论（tensor calibration theory） 张量校正理论可看成为构成分析化学第二层次的基础理论的重要组成部分。简言之，张量校正理论就是分析化学中的定量分析理论，它将常用的经典分析化学中的灵敏度、选择性、信噪比、检测下限以及结果可靠性评价等已为广大分析化学工作者广泛接受的基本概念，拓广至矢量甚至张量数据，将为分析工作者针对不同分析体系，选择正确定量校正方法、评价分析结果的可靠性提供理论依据。

直接校正方法（direct calibration methods） 直接校正方法是多元校正中一类用标准物量测谱直接对混合物量测谱进行校正的方法。根据所采用的不同数学方法，可分为多元线性回归方法（MLR），Kalman 滤波方法（KF）和加权最小二乘回归法（WLSR）。

植入误差（imbedded errors） 在因子分析中，量测误差被分为两项，即所谓的植入误差（IE）和抽出误差（XE）。其中植入误差（IE）为存在于主因子轴的误差，属于一种用因子分析方法无法剔除的一部分误差。同时，它与抽出误差（XE）和量测误差即真实误差（RE）的关系，可由一勾股定理的方式表出，即

$$(RE)^2=(IE)^2+(XE)^2$$

自适应 Kalman 滤波法（adaptive Kalman filtering） 自适应 Kalman 滤波法是一种灰色分析的校正方法。一般的 Kalman 滤波都是用不变的 $r(k)$ 来进行计算，而自适应 Kalman 滤波方法就是通过在滤波过程中调节 $r(k)$ 来抵制模型错误，从而达到在有未知干扰的存在下估计待测物浓度的目的。必须指出的是，对于灰色分析体系，自适应 Kalman 滤波法得到的也只是一个可能解，而不是唯一解。

主成分回归法（principal component regression） 主成分回归法实际上可以看成是对 P-矩阵法的一种改进。它不但可尽可能多地保持有用信息，即保留所有的线性量测点，而且还保持了 P-矩阵法的既可进行实验设计又可一步求解的优点。它采用多元统计中的主成分分析方法，先对混合物量测矩阵 $\boldsymbol{Y}$ 直接进行分解，然后只取其中的主成分来进行回归分析，故有主成分回归之称。

诊断方法（diagnostic methods） 在统计学中，一般用两种方法来处理含奇异点和非正态分布的数据，即诊断方法和稳健估计方法。这两种方法虽处理顺序完全相反，但却殊途同归，具有相同目的。诊断方法是着重于首先找出奇异点，在剔除这些奇异点后继用经典估计

方法处理，以达到最终稳健的效果。

最小中位平方法 (least median of squares) 最小中位平方法是一种具有高崩溃点的稳健估计回归方法。它的目标函数是

$$\mathrm{Min\ median}(r_i^2)$$

在这里 $\mathrm{median}(r_i^2)$ 表示残差平方的中位数。最小中位平方估计法的崩溃点可以达到50%，这是稳健估计方法的可能达到的最高值。然而，LMS 估计的收敛速率较慢。一般说来，LMS 方法的计算时间与量测样本的点数(n)的立方成正比。

最小修剪平方法 (least trimmed squares) 最小修剪平方法是一种稳健回归分析方法，为克服 LMS 估计的较慢收敛速率之不足，Rousseeuw 提出了最小方差修剪估计，其目标函数为

$$\mathrm{Min}\sum_{i=1}^{h}\sum(r_i^2)_{i:n}$$

在这里 $(r_i^2)_{1:n} < (r_i^2)_{2:n} < \cdots < (r_i^2)_{n:n}$。从上式可以看出，此法与最小二乘类似，不同的地方只是 LTS 估计只取了一部分残差较小的来进行回归（即从 $(r_i^2)_{1:n}$ 到 $(r_i^2)_{h:n}$）。当取 h 为 n 的一半时，LTS 估计的崩溃点也可达到50%，但是它的收敛速度明显加快。

最小一乘法 (least absolute residuals) 最小一乘法首先是由 Edgeworth 在1887年提出，其目标函数为 $\mathrm{Min}\sum|r_i|$，以减小最小二乘由于对残差进行平方而增大异常点对拟合的影响，从而达到稳健的效果。由于绝对值的计算不便于微分，最小一乘很难得到像最小二乘那样的简洁解析解，一般需采用优化方法求解。

最小体积椭球估计 (minimum volume ellipsoid estimates, MVE) 最小体积椭球估计是一种稳健诊断方法。它通过迭代过程不断剔除奇异点，而求得一个最小体积的椭球来表示服从多元正态分布的数据点的主体。

秩消失因子分析法 (rank annihilation factor analysis) 秩消失因子分析法是针对二维荧光谱的定量分析而提出的。该方法的基本原理就是利用纯物种二维荧光光谱是一个双线性矩阵，且其秩为一的特点来进行的。此方法的基本思路为：设已测得被分析物的纯物种二维谱，将其作为标准，进而对含未知干扰的混合体系的二维荧光光谱来进行消去该标准的运算。设未进行消去运算前，含未知干扰的混合体系的量测矩阵的秩为 k，那么，当被消去的标准正好等于其存在于混合体系的该物种的浓度大小时，未知干扰的混合体系的量测矩阵的秩就会减少1而成为 $(k-1)$，据此即可对该被分析物进行定量。

自模式曲线分辨法 (self-modeling curve resolution) 自模式曲线分辨法是一种基于矩阵数据的多元分辨方法，主要基于主成分分析 (PCA)，其基本假设只有两条：① 混合物量测谱符合线性加合性；② 波谱量测值只能为正值。因这两条假设要求条件对一般波谱仪器所产生的数据皆可成立，故其适用性较广。该法最早由 Lawton 和 Sylvestre 提出，他们以混合物体系的纯物种光谱解析为目标，以得到它们的可行解域，后被拓广至三组分或三组分以上的混合体系。随着组分数目的增加，纯组分光谱的可行解域确定的困难显著增加。

直观推导式演进特征投影法 (heuristic evolving latent projections, HELP) 直观推导式演进特征投影法是采用一种逐步的推导式方式来解析二维色谱数据，与自模式曲线分辨法仅基于全局主成分分析的思路不同，它主要基于局部主成分分析，以尽量利用色谱的分离特性来获得各子矩阵的有用信息，其中最重要的是所谓选择性信息，在此基础上，它试图根据色谱流出数据特性来获得在色谱保留时间方向上的所谓"秩图"，实际上就是色谱保留时间方

向上组分分布图；然后根据秩图和所得的选择性信息来对重叠的色谱或光谱进行满秩分解。整个 HELP 算法将有以下几个步骤来完成：

① 背景的检测和扣除。利用演进特征投影图对要分辨的数据进行分析，如果零组分区的光谱投影点没有集中于原点，则表示存在背景，然后利用零组分区的信息来扣除背景。

② 数据预处理与体系组分数的确定。对于存在有不等性方差噪声的数据，在用特征结构跟踪分析（ETA）来确定体系的秩图之前，有必要进行预处理。另外，在决定体系的组分数时，同样有必要利用演进特征投影图，因此图对各组分的依次流出将给出很重要的信息，以帮助证实体系的组分数。

③ 选择性信息和零浓度区的确定。同时利用演进特征投影图和特征结构跟踪分析所得的秩图来确定选择性区域以及该组分的零浓度区域。

④ 纯组分光谱和色谱和满秩分辨。利用满秩分辨技术对已知选择性信息和零浓度区域的组分进行分辨，求得其转换矢量以获得该组分的纯光谱和纯色谱。对于组分数大于 3 的多组分体系，可采用组分剥离技术来逐步进行。

⑤ 对分辨所得结果进行证实。将分辨所得的各纯光谱与纯色谱乘积的线性加合结果与实际数据进行比较，以进一步确证所得结果的可靠性。

正交投影分辨法（orthogonal projection resolution，OPR） 正交投影分辨算法是一种基于矩阵数据的多元分辨方法，它主要是基于 Lorber 提出的正交投影的基本思路而提出，在结构上与 Malinowski 提出的窗口因子分析法（WFA）类似，它也是利用已知的体系中含 n 1 个物种的窗口来对第 n 个物种进行正交投影以求得其色谱流出信息。只是采用投影矩阵的方法来实现而已。

第四节　分析检测理论与信号处理

粗糙惩罚平滑法（roughness penalty smoothing method） 粗糙惩罚平滑法是一种非参数估计的方法。其基本思路是在最小二乘估计的基础上再加上一个粗糙惩罚项来构成其目标函数。粗糙惩罚平滑法一方面注意到了拟合的作用（此点优于简单平均），另一方面，又注意到了测量数据的粗糙性，越是粗糙和不光滑的测量数据中噪音的成分就越大，受到的惩罚亦越大。通过交叉校验（cross-validation）来决定惩罚程度，可使目标函数对拟合和噪声惩罚达到最好的平衡，以最大限度保证分析信号既不失真又能消除噪声，取得最好的平滑效果。

窗口移动平均法（window movingaverage method） 对于一个含噪声的波谱或电化学谱，若以对其中第 i 个点及其前后各 m 个数据点进行平均所得之值来替代第 i 个点的测量值，在此，$(2m+1)$ 就是窗口的宽度。然后依次移动此窗口，直至将整个波谱量测数据点全用窗口平均值来替代，这种简单的平滑法称之为窗口移动平均法。

窗口移动多项式最小二乘拟合平滑法（Window moving smooth method based on polynomial least squares） 窗口移动多项式最小二乘拟合平滑法由 Savitzky 与 Golay 提出。此法不是采用简单平均运算，而是在窗口移动运算中引入多项式最小二乘拟合，故该法可做到既能去噪提高分析信号的信噪比，又可较好地保持分析信号（波谱）中的有用信息。此法虽早在 20 世纪 60 年代提出，但至今仍在分析化学信号预处理中频繁使用。

窗口移动中位数平滑（window moving median smoothing） 由于平均数的计算和最小二乘估计是非稳健的（non-robust），所以对于一些存在有奇异点的波谱，窗口移动平均平滑法和窗口移动多项式最小二乘拟合平滑法皆难于给出满意的结果。如果用稳健的中位数来代

替平均数，就构成了窗口移动中位数平滑法。

窗口 Fourier 变换（windowed fourier transform，WFT） Fourier变换将整个时域信号变换为频域信号，而失去了时域中的分辨率。为此出现了窗口 Fourier 变换（WET）或短时 Fourier 变换（STFT），它通过对信号适当预加窗以得到一个局部谱，它等价于取出信号在分析点 $t=t'$ 附近的一个切片。所以，窗口 Fourier 变换直接是信号 $x(t)$ 在“分析时间” t' 附近的“局部谱”。

其中 rank（$\boldsymbol{M}|\boldsymbol{N}$）表示在 $\boldsymbol{M}$ 中除去 N 的秩。此二阶张量校正模型的选择性 $SEL_N=\|\boldsymbol{NAS}_N\|/\|\boldsymbol{N}\|=1$，说明此类模型具有完全选择性，故可在未知干扰存在下直接对待求物种进行定量解析。

尺度（scaling） 在分析信号的小波变换处理中，一般实际使用的是离散小波变换，离散小波变换定义式为

$$h_{a,b}(t)=2^{-j/2}h(2^{-j}t-k)$$

其中，j，k 为任意整数。如有一离散信号 $x(t_i)=x_i$（$i=1$，…，n），在不同的 j，k 下，其离散小波变换也就相当于

$$\begin{aligned} WT_x(j,k) &= \textstyle\sum_{i=1}^{n} 2^{-j/2}h(2^{-j}t_i-k)x_i \\ &= \langle x_i, 2^{-j/2}h(2^{-j}t-k)\rangle \end{aligned}$$

式中，⟨·，·⟩表示内积，也就是投影。此式说明小波变换就相当于一种在小波基函数上的投影变换。在此，不同的 j 就代表不同的尺度（分辨率），小波变换正是通过不同的 j 来调节不同的分辨率的。

尺度函数（scaling function） 小波基函数的表达式为

$$h(t)=\textstyle\sum_k \alpha_k \sqrt{2}h(2t-k)$$

此式说明，$h(t)$ 就是由 $h(2t-k)$ 的线性组合而成。此式中的 $h(t)$ 称为尺度函数，α_k 则称为尺度函数系数，或尺度滤波（scaling filter）、尺度向量（scaling vector）。

短时 Fourier 变换（short time Fourier transform，STFT） 短时 Fourier 变换是窗口 Fourier 变换的另一个名称。

多分辨分析（multiresolution analysis） 小波变换的应用都是通过 Mallat 提出的多分辨信号分解（MRSD）或塔式算法来实现的，其算法在小波变换中的地位就与快速 Fourier 变换（FFT）在 Fourier 变换中的地位一样重要。

多分辨信号分解（multiresolution signal decomposition，MRSD） 多分辨信号分解（MRSD）或塔式算法是由 Mallat 提出的一种著名的算法，它可将分析信号矢量按不同的分辨率来进行分解和压缩，在小波分析中占有十分重要的地位。

Fourier 变换（Fourier transformation） Fourier 变换是一种通过以三角函数为其基函数的数学变换方法，它可将各种不同分析信号从时域变换到频域，在分析信号处理中，它既可用于除噪，解卷积，还可用于光谱波谱分析中的快速频域变换，是一种十分有效的化学计量学变换方法。

光谱多重性效益（multiplexing advantage in spectroscopy） 所谓光谱多重性效益是将单通道光路通过一种编码方式来增加光谱量测信噪比的方法，如用 Hadamard 编码使光谱分析信噪比提高 $(N+1)/2N^{1/2}$ 倍，在 N 很大时，约为 $N^{1/2}/2$ 倍。这种光谱多重性效益，又称 Fellgett 效益。用 Hadamard 变换设计的红外光谱与常规红外光谱比较，有改善信噪比的效果。

Hadamard 变换（Hadamard transform） 各元素的绝对值均为 1 的矩阵，其行列式为 1，且其逆矩阵 $H^{-1}=\frac{1}{N}H$，则该矩阵称为 Hadamard 矩阵，记为 H，如一个分析信号矢量 x，经变换后得 $m=Hx$。这种变换称为 Hadamard 变换。在光谱分析中，传统的扫描方法测试光谱，每次测一个波长的光强度，如改为同时测定多波长的光强度，有 N 个波长，进行 N 次不同的组合测试，即进行 Hadamard 变换，则可减少测试误差，或改善光谱量测的信噪比。

卷积（convolution） 卷积是统计学中的一个概念，在信号处理中有着十分重要的意义。与窗口移动平滑法类似，其离散表达式为

$$g[x(i)] = \sum_{i=-m}^{m} f(x)h[x(i) - x]$$

其连续形式表出则变为

$$g[x(i)] = \int_{-\infty}^{+\infty} f(x)h[x(i) - x]\mathrm{d}x$$

式中，$g(x(i))$称为函数 $f(x(i))$与 $h(x(i))$的卷积，记为 $f(y) * h(y)$。卷积亦称褶积，取h 函数相对坐标折迭之意。卷积运算实质上也是一种变换运算。

快速 Fourier 变换（fast Fourier transform，FFT） 快速 Fourier 变换（FFT）的基本思路是将离散 Fourier 变换式进行分解，以减少运算。在 N 是以 2 为底的幂时（即 $N=2^a$，a 为正整数），这种分解十分方便，如再用二进制来替代复数运算，则可使 Fourier 变换将乘法由 N^2 次降到$\frac{Na}{2}$次。在此，$N=2^a$，亦即 $a=\lg N$. 如 $N-1024=2^{10}$，计算量由原先 1024^2 降到 $\frac{1024\times10}{2}$，从而大大加速了 Fourier 变换速度，故有快速 Fourier 变换之称。

离散 Fourier 变换（discrete Fourier transform，DFT） 在Fourier 变换的实际应用中，一般用的是离散 Fourier 变换，离散 Fourier 变换就是将连续的分析信号 $x(t)$离散化，即对分析信号 $x(t)$在均匀时间间隔 Δ 进行记录，可得序列样本值：

$$x_n=x(n\Delta) \qquad n=\cdots,-3,-2,-1,0,1,2,3,\cdots$$

设采集 N 个序贯样本值：

$$x_k=x(t_k) \quad t_k=k\Delta \qquad k=0,1,2,\cdots,n-1$$

就可以离散值之和来近似表达连续 Fourier 变换，即

$$X(f_n) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-j2\pi f_n t}\mathrm{d}t = \Delta\sum_{k=0}^{N-1} X_k \mathrm{e}^{-j2\pi f_n t_k} = \Delta\sum_{k=0}^{N-1} X_k \mathrm{e}^{-j2\pi\left(\frac{n}{N\delta}\right)K\delta}$$
$$= \Delta\sum_{k=0}^{N-1} X_k \mathrm{e}^{-j2\pi k\left(\frac{n}{N}\right)}$$

这就是所谓离散 Fourier 变换（DFT），亦即

$$X(n) = \sum_{k=0}^{n-1} x(k)\mathrm{e}^{-j2\pi k\left(\frac{n}{N}\right)}$$

离散小波变换（discrete wavelet transformation） 在分析信号的小波变换处理中，一般实际使用的是离散小波变换，将小波变换定义式中的小波基函数 $h_{a,b}(t)=(a^{-1/2})h[(t-b)/a]$ 的平移和伸缩或尺度参数离散化，即得

$$h_{a,b}(t)=2^{j/2}h(2^j t-k)$$

其中，j，k 为任意整数。

如有一离散分析信号 $x(t_i)=x_i(i=1,\cdots,n)$，在不同的 j，k 下，其离散小波变换也就相当于

$$WT_x(j,k)=\sum_i 2^{-j/2}h(2^{-j}t_i-k)x_i$$
$$=\langle x_i,2^{-j/2}h(2^{-j}t_i-k)\rangle$$

式中，〈·，·〉表示内积，也就是投影，此式说明小波变换就相当于一种在小波基函数上的投影变换，不同的 j, k 代表不同的分辨率（尺度）和不同时域（平移），这样的小波变换就称为离散小波变换。

平滑方法（smoothing method） 平滑方法是分析信号预处理中一种常用的去噪以提高信噪比的方法，在分析化学的信号预处理中得到了相当广泛的应用。在化学计量学中的平滑方法包括最简单的窗口移动平均法，窗口移动多项式最小二乘拟合法、稳健中位数法、粗糙惩罚平滑法。

塔式算法（pyramid algorithm） 在小波变换中，塔式算法是多分辨信号分解（MRSD）算法的别名，其算法在小波变换中的地位与快速 Fourier 变换（FFT）在 Fourier 变换中的地位一样重要。

小波函数（wavelet function） 小波变换函数的基本定义为

$$WT_x(a,b)=\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty}h\left(\frac{t-b}{a}\right)x(t)\mathrm{d}t$$

从此式可以看出，它与窗口 Fourier 变换很相似，不同之点只在于小波变换少了三角函数变换（$e^{-2\pi ft}$）一项，但增加了一个伸缩参数 a，对于大的伸缩参数 a，基函数变成展宽的原像小波，是一个低频函数；而对于小的 a，基函数则成为缩小的小波，它是一个短的高频函数。正是这一伸缩参数，使小波具有多分辨的性质。小波变换的另一参数为平移参数 b，这就使它也同样具有前述窗口 Fourier 变换可得到时域（或空间域）“局部谱”的特性。另外，小波变换不像 Fourier 变换只采用三角函数为其基底函数，其基函数 $h(\cdot)$是可改变而且可以相互正交的。不同的基函数 $h(\cdot)$对应于不同的小波变换，最著名和常用的小波基函数有 Haar 基、Daubechies 基、Coiflet 基和 B-样条基等。正是因为有了这样一些特点，使小波变换成为了一种比 Fourier 变化更灵活和更多样化的信号处理的数学新工具。

小波多分辨变换（wavelet multiresolution transformation） 小波变换的最显著特性就是它的多分辨特点。用多分辨函数来逼近任一函数的基本概念是由 Meyer 和 Mallat 提出的，这相当于我们用显微镜来观察一个微生物，如果我们想要该微生物的总貌，则需将其调至较大视野（低频），而不在意该微生物各部位的细节；反之，如我们想了解某一局部部位，则需聚焦以高分辨来观察细节。小波多分辨分析正是根据这一基本思路来进行设计的。在小波变换中，有一个伸缩参数 a，对于大的伸缩参数 a，基函数变成展宽的原像小波，是一个低频函数；而对于小的 a，基函数则成为缩小的小波，它是一个短的高频函数。正是这一伸缩参数，使小波变换具有多分辨的性质。

第五节　化学模式识别

标准化（normalization） 标准化是一种数据预处理的方法，对不同的仪器可有不同的方法，如在色谱数据中常采用面积归一化，以减小进样误差的影响，而对于质谱数据则常采用最大丰度归一化的方法。

Bayes 判别法（Bayes discrimination） Bayes判别法建立在条件概率和统计判决理论的 Bayes 定理之上。其核心就是找到一个划分界或一个判别界，使得在此划分界的基础上计算样本被误判的平均损失达到最小的一种模式识别方法。

范围标度化（range scaling） 范围标度化是模式识别中的一个数据预处理方法，它可由下式表示：

$$x_{ij,\text{new}}=\frac{(x_{ij,\text{old}}-x_{ij,\text{最小}})}{(x_{ij,\text{最大}}-x_{ij,\text{最小}})}$$

在此，$x_{ij,\text{old}}$代表第 i 个样本的原有的第 j 分量，$x_{ij,\text{new}}$代表第 i 个样本经换算后的第 j 分量，$x_{ij,\text{最大}}$代表原有的第 j 分量的最大值，$x_{ij,\text{最小}}$则代表原有的第 j 分量的最小值。其目的在于将不同测量方法所得的变量值统一起来，都变成到 0～1 范围之内。

Fisher 判别法（Fisher discrimination） Fisher判别法的基本思路是先计算出不同类别样本的组间方差和组内方差，而后构造一个组间方差和组内方差的比值。Fisher 判别法是一个可使得组间方差和组内方差的比值达到最大的判别方法。

化学模式识别（chemical pattern recognition） 化学模式识别是化学计量学研究中的一个十分重要的内容，它从化学量测数据出发，进一步揭示物质的隐含性质，为化学家提供了十分有用的决策性信息，模式识别的计算过程总的说来可有如下步骤：根据实验得来的一批训练点，参照化学（或物理）模型或经验规律提出一批特征量，然后进行进一步特征抽取，以求得合适的特征量，张成模式空间或特征空间，必要时，对数据进行预处理。预处理后，即可通过模式识别算法进行训练和分类，然后根据训练（或称学习）分类所得的判据对未知样本进行判别（或称计算机预报）。化学模式识别方法，一般可分为所谓的有监督的模式识别方法（判别分析）和无监督的模式识别方法（聚类分析）。

句法模式识别技术（syntactic pattern recognition） 句法模式识别技术是一种有别于数值模式识别的方法，它主要依靠逻辑而不是数值的方法来进行模式识别。这种方法还很少见用于来解决化学问题。

距离判别法（distance discrimination method） 距离判别法是一种通过距离来进行判别分析的方法。对给定的一个样本，要判别它来自哪个母体，一个最直观的方法就是计算样本到各类 G_1，G_2，…，G_k 的距离 $d(\boldsymbol{x}, G_1)$，$d(\boldsymbol{x}, G_2)$，…$d(\boldsymbol{x}, G_k)$，然后根据样本 $\boldsymbol{x}$ 到它们的距离来确定样本的归宿，这就构成了距离判别的基本思路。

聚类分析法（clustering analysis） 聚类分析法是一种无监督的模式识别方法。它是多元统计中的一种实用性很强的方法，其主要思路就是利用同类样本应彼此相似，即常说的“物以类聚”，相类似的样本在多维空间中彼此的距离应小些，而不相似的样本在多维空间中彼此的距离应大些。聚类分析即为如何使相似的样本“聚”在一起，从而达到分类的目的。一般说来，聚类分析方法可分为凝聚法与分离法两类，凝聚法开始时将每个样本认作一个类，然后由这些类逐步凝聚成较大的类；而分割法则与之相反，开始时将所有的样本认作一个大类，再逐步分割为较小的类。

基于全局寻优的聚类法（clustering methods based on global optimization） 聚类分析实际上可转化成为一个优化问题，但由于聚类目标函数的严重非线性，用传统的局部优化方法难以找到全局最优解，因此需用全局优化方法。若将 100 个样本分为 2 类，用穷举法需计算高达$2^{99}-1$种划分，以确定最优解，且用其他确定性的全局优化方法也因计算量过大而难以实现。随机性的全局优化方法在处理上述问题上优于确定性全局优化方法，遗传算法和模拟退火作为随机性的全局优化方法已开始引人化学聚类分析，它们的共同之处是将聚类分析构造成非连续的组合优化问题，即通过寻找各个样本对各类的划分而求得最优目标函数，此类方法计算量巨大，需消耗很长时间才能找到最优点。也可构建连续型优化模型，即通过找出

各类中心的正确位置而达到正确聚类，使所有点到其所属类中心距离的和最小，从而避免求解组合优化问题，减少了计算量，提高了优化效率。

***K*-最近邻法**（*K*-nearest neighbor classification method，*K*-NN） *K*-最近邻法是一种直接以模式识别的基本假设——同类样本在模式空间相互较靠近——为依据的分类方法。它计算在最近邻域中 k 个已知样本到未知的待判别样本的距离，而后决定该样本属于哪一类的非参数模式识别法。对于线性不可分体系，此法仍可适用。

***K*-均值聚类法**（*K*-means clustering method） *K*-均值聚类法是一种较新的聚类分析方法。它的基本思路是先人为地定出一个分类数目 k，并任取 k 个点作为初始凝聚点，然后，逐个将别的样本引入，引进一个样本就计算它与 k 个类的距离（与重心的距离），选定距离最小者将其并入该类，再重新计算各类的重心，并以该重心作为新的凝聚点，直至每个样本都有了各自的归类。此方法在化学计量学中得到较广泛地应用。

可变类平均法（variable class average method） 可变类平均法是系统聚类法的一种方法。因在类平均法的递推公式中没有反映 D_{pq}^2 的影响，故有人建议将它递推公式改为

$$D_{ir}^2=[(n_p/n_r)(1-\beta)D_{ip}^2+(n_q/n_r)(1-\beta)D_{iq}^2+\beta D_{pq}^2]$$

其中，$\beta<1$，对应于这个递推公式的系统聚类法叫做可变类平均法。

可变法（variable method） 如将可变类平均法的递推公式改为

$$D_{ir}^2=[(1-\beta)/2](D_{ip}^2+D_{iq}^2)+\beta D_{pq}^2$$

其中，$\beta<1$，那么，对应于这个递推公式的系统聚类法叫做可变法。

类平均法（class average method） 类平均法是系统聚类法中的一种。重心虽是一类点集的很好的代表，但未充分利用各样本的信息，有人建议将两类之间的距离定义为这两类元素所有样本两两之间的平均平方距离，即

$$D_{pq}^2=(1/n_pn_q)\sum_{\boldsymbol{x}_i\in G_p,\boldsymbol{x}_j\in G_q}\sum d_{x_ix_j}^2$$

在此，n_p 和 n_q 分别为类 G_p 与 G_q 中的样本数。此种方法的递推公式为

$$D_{ir}^2=[(n_p/n_r)D_{ip}^2+(n_q/n_r)D_{iq}^2]$$

有人认为类平均法是系统聚类法中比较好的方法之一。

马氏距离（Mahalanobis distance） 在统计学和化学计量学中，马氏距离是一个常用的诊断工具，它的定义式为

$$MD_i=(\boldsymbol{x}_i-\boldsymbol{x})\boldsymbol{C}^{-1}(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^t$$

在这里 $\boldsymbol{C}$ 是 $\boldsymbol{X}$ 矩阵的协方差阵，而 $\bar{\boldsymbol{x}}$ 则是 $\boldsymbol{x}_i(i=1,2,\cdots n)$ 的均值矢量。这种诊断方法对于单个的奇异点或杠杆点都有一定作用，但是，由于杠杆点具有很强的相互掩蔽效应，以致造成很多假象而掩蔽了真正的奇异点和杠杆点。出现这种现象的原由是马氏距离是建立在不稳健的经典协方差矩阵之上。

模式空间（pattern space） 任何一个化学样本或化工过程 $\boldsymbol{x}_i$ 可用一组参量（矢量）来表征，即 $\boldsymbol{x}_i=(x_{i1}, x_{i2}, \cdots, x_{in})$，在化学模式识别的文献中，这种参量值又称特征。在许多情况下，这些参量实际上就是原始化学量测数据。这样的一组 n 个参量就构成了化学模式识别空间的一个点，一个 n 维的化学模式。由这样的化学模式所构成的 n 维空间，称化学模式空间，简称模式空间。

模式空间的距离（distance in pattern space） 化学模式识别的基本假设是相似的样本

在模式空间处于相近的位置，如何度量两个样本在模式空间的远近？可应用在几何学中的距离的概念，它在模式识别中是作为与“相似性”相反的“相异性”的度量使用的，任何满足下述三个条件的“相异性”均可定义为模式空间的距离：

① $D_{ij} \geqslant 0$；如 $\boldsymbol{x}_i = \boldsymbol{x}_j$，$D_{ij} = 0$

② $D_{ij} = D_{ji}$

③ $D_{ik} + D_{jk} \geqslant D_{ij}$

条件①和条件②表明距离应取正值且是对称的，条件③是作为“度量”距离与“非度量”距离的差异。在欧氏空间中，两点间的直线距离恒小于或等于其他路径的长度。

模糊聚类法（fuzzy clustering method） 在聚类问题上可采用硬分类和模糊分类两种方法确定目标函数。所谓硬分类的结果是每个样本将绝对地划归某一类，即每个样本仅能属于一个类；而模糊分类的结果是每个样本可在不同程度上或多或少地属于各个不同的类。

判别分析法（discrimination method） 判别分析法是多元统计中的一种分类方法，它通过已知类别数据而求出不同类别的数据点的判决面，进而可对未知类别数据点进行分类，是一类有监督的化学模式识别法。

偏最小二乘的投影判别法（projection discrimination based on PLS） 因为偏最小二乘方法实质上是一种特征变量方法，所以，如果将样本点投影到偏最小二乘方法求得特征变量轴上，就可将高维空间的点集投影到低维空间中来用肉眼观察高维空间的点集的数据结构，从而达到对它们进行分类的效果。这样的方法就称为偏最小二乘的投影判别法，它在化学计量学中得到了相当广泛的应用。

人工神经网络判别法（discrimination based on artificial neural networks） 人工神经网络用于化学模式识别的基本思路与一般的判别分析方法是相同的，它也是通过对已知类别的训练集的拟合建立模型来进行分类与预测的。它与非线性校正方法的区别仅在于校正方法的拟合为连续变化函数，而化学模式识别则是以 0，1 或 −1、+1 为其输出来进行拟合建模的。由于人工神经网络具有非线性映照的能力，近年来在化学模式识别中得到了相当广泛的应用。

识别率（recognition ratio） 识别率是一种检测所得判别模型的判别能力的标准，它表示在训练中所得的正确判别率。一般说来，预测率对模型好坏的判别比识别率更重要些。

数据预处理（data preprocessing） 在化学模式识别中一般都需对数据进行预处理。例如，各特征变量取了不同量纲的数值时，如果有的变量取的绝对值很大，而另外一些变量的绝对值很小，则在分类过程中不利于获得最佳结果。因此应当用标度化方法使各变量的变化幅度处于同一水平上，它们包括：范围标度化、自标度化、标准化、变换法和组合法。对于不同的情况可以采用不同的预处理方法，这不但需要正确理解各种预处理方法的物理意义，同时也必须结合不同问题的具体化学经验来对原始数据进行预处理，不可统而论之。

势函数判别法（potential function discrimination） 势函数分类法实际上是 K-最近邻法（K-NN 法）的一种很有效的拓广变种。它把每一个已知种类样本的代表点都看成某种“势场”的源，例如一个电场源即一个点电荷。对于未知点属于哪一类的判断基于这样一个原则：未知点属于在该点有最大势场强度的那一类。一般说来，势函数分类法分界面为复杂的不连续的曲面。

SIMCA 分类法（SIMCA classification） SIMCA分类法，又称简易分类算法（simple classification algorithm，SIMCA）或簇类的独立软模式分类法（Soft Independent modeling of class analogy，SIMCA）。它是基于主成分分析的一类分类法，在化学计量学中得到相当广泛

的应用。

特征投影的模式识别方法 (pattern recognition by latent projections) 通过特征投影的方式，在尽可能减少原 d 维空间中分类信息的丢失的前提下，将高维模式空间的样本点在二维或三维空间中显示出来，继借人眼识别模式分布情况，直接所视分类图景，这就是所谓的基于投影的模式识别方法的主要思路。基于投影的模式识别方法有很多，其中最重要的有主成分分析的投影判别法（亦称 Karhunen-Loeve 变换，K L 分解法）、SIMCA 方法、基于偏最小二乘分解的特征投影法、非线性映照投影法等，因其具有图形显示而具有直观性，很容易为化学家所理解，故它们在化学模式识别中得到相当广泛的应用。

特征抽取方法 (feature extraction) 在进行模式识别之前进行特征抽取，选取合适的特征量，以张成合适的特征空间，是化学模式识别工作成败的一个关键。特征量的选取标准可分为化学(或物理)的和数学的两方面。化学(或物理)的选取标准基于对所处理的信息的本质或主要因子的理解。这方面因其依赖具体问题的实际情况，无法指出普遍有效的方法。由于对所处理的信息了解不深，而且往往有许多因素之间的相关关系及程度亦不能肯定，因此人们通常是先根据化学(或物理)的选取标准尽可能把一切可能有关，又易于获得的因素(特征量)都提出来，然后借助于数学方法，筛选出对分类作用大的特征量，张成合适的特征空间供分析使用。在实际的计算中，人们总是力图抛弃那些对分类作用不大的特征量，使特征量的数目(在保证良好的分类效果的前提下)减到最小。这样的一些方法就称为特征抽取方法。

无监督的模式识别 (unsupervised pattern recognition) 无监督的模式识别是一种事先没有已知类别的分类方法。聚类分析就是无监督的模式识别法的代表，其主要思路就是利用同类样本应彼此相似，即常说的“物以类聚”，相类似的样本在多维空间中彼此的距离应小些，而相似的样本在多维空间中彼此的距离应大些。聚类分析即为如何使相似的样本“聚”在一起，从而达到分类的目的。

谱系聚类法 (hierarchical clustering method) 谱系聚类法亦称系统聚类法，是一种很常用的凝聚法方法，其基本思想是：先将 n 个样本各自看成一类，然后规定样本之间的距离和类与类之间的距离。开始，因每个样本自成一类，类与类之间的距离与样本之间的距离是等价的，选择距离最小的一对并成一个新类，计算新类与其他类的距离，再将距离最小的两类并成一类，这样每次减少一类，直至所有的样本都成为一类为止。

线性学习机 (linear learning machine) 线性学习机是一种简单的模式识别方法。线性学习机又称线性判别函数法。线性判别函数法的目标是找到多个矢量 w 来对模式空间进行划分，如果它们是线性可分的，则总可找到多个矢量 w，使得模式空间的不同类样本可分。

一分为二法 (one divided in two method) 一分为二法是一种聚类分析方法。它的分类思想是分解法，即是使类由少到多，开始全体样本全为一类，然后分成两类、三类、……直至所有的样本各自成一类。一分为二法就是这样一种分解方法。其基本思路是将某一类分解成为两子类的方法，然后对其子类又可一分为二，在不断的分解过程中，使得类间方差保持最大。

有监督的模式识别 (supervised pattern recognition) 有监督的模式识别的分类方法很多，这类方法的基本思路都是用一组已知类别的样本作为训练集，即用已知的样本进行训练，让计算机向这些已知样本“学习”，这种求取分类器的模式识别方法称为“有监督的学习”或“有老师的学习”。这里训练集就是老师，并由这个训练集得到判别模型。这类方法一般可分为参数法和非参数法两类，其中参数法一般由统计学家提出，其判别效果的好坏依赖于样本是否符合假设的统计分布，而非参数法多由实验科学家或计算机模式识别专家提出，对样本

分布没有特殊要求。几种最常用的方法，包括几种经典的参数方法，如距离判别分析法、Fisher判别分析法、Beayes 判别分析法、逐步判别分析法，以及几种非参数方法，如线性学习机、K邻域判别法、势函数判别法、人工神经网络判别法。

有监督的学习（supervised learning） 有监督的学习是表示模式识别中的一个学习过程。一般是用一组已知类别的样本作为训练集，即用已知的样本进行训练，让计算机向这些已知样本"学习"，这种求取分类器的模式识别方法称为"有监督的学习"或"有老师的学习"，这里训练集就是老师。并由这个训练集得到判别模型。

预测率（prediction ratio） 与识别率一样，预测率也是一种用来检验模式识别方法效果的指标。为检测所得判别模型的判别能力，常用另外一组已知类别的样本组成测试集。在训练中所得的正确判别率称为识别率，而用测试集所得的正确判别率则称为预测率，一般说来，预测率对模型好坏的判别比识别率更重要些。

最短距离法（minimum distance method） 最短距离法是系统聚类分析方法的一种，其主要思路是定义类 G_i 与 G_j 之间的距离为

$$D_{ij}=\min\{d_{kl}\}$$
$$\boldsymbol{x}_k\in G_i \quad , \quad \boldsymbol{x}_l\in G_j$$

式中，d_{kl}是样本 $\boldsymbol{x}_k$ 与 $\boldsymbol{x}_l$ 的距离，即类与类之间的距离为两类之间最近的样本之间的距离。

最长距离法（maximum distance method） 最长距离法也是系统聚类分析方法的一种，其主要思路是定义类 G_i 与 G_j 之间的距离为

$$D_{ij}=\max\{d_{kl}\}$$
$$\boldsymbol{x}_k\in G_i \quad , \quad \boldsymbol{x}_l\in G_j$$

式中，d_{kl}是样本 $\boldsymbol{x}_k$ 与 $\boldsymbol{x}_l$ 的距离，即类与类之间的距离为两类之间最远的样本之间的距离。

最小生成树（minmun spanning tree） 最小生成树是一种基于图论的聚类分析方法。在图论中，没有回路的联结图叫做树。如果树包含了 n 个点，则称这个树为联结图的支撑树，树的所有边的长度之和叫做树的重量。在联结图中具有最轻重量的支撑树叫做最小生成树，亦称最小支撑树。最小生成树是按 Kruskal 算法找出的，这个算法十分简单，要求将权值最小的边绘人树中，但不得与已有的边形成回路。其算法如下：

① 先计算各点间的距离；

② 连接最小距离的两点，然后连接次小距离的两点；

③ 继续连接所剩下点的最小距离的两点，并检查是否形成回路，如形成回路则取消连接，否则继续连接，直至所有的点都被连接；

④ 对连接所得的树进行检查，找到最大距离的边，将其割断，即得两类，如此继续分割，直至类数已达到所要分的类数。

自标度化（autoscaling） 自标度化是化学模式识别中的数据预处理的一种方法，它的定义为

$$x_{ij,\text{new}}=(x_{ij,\text{old}}-m_j)/V_j$$

式中，V_j 为变量 j 的方差，m_j 为变量 j 的均值。自标度法是一个很好的方法，经此变换后，各变量在分类过程中影响程度相同。

主成分投影判别法（projection discrimination based on PCA） 主成分分析法实质上是一种特征变量方法，所以，如果将样本点投影到主成分的特征变量轴上，就可将高维空间的点集投影到低维空间中来；从而可用肉眼观察高维空间的点集的数据结构，从而达到对它们

进行分类的效果。这样的方法就称为主成分投影判别法，它在化学计量学中得到了相当广泛的应用。SIMCA 分类法就是基于主成分投影判别法而提出的。

中间距离法（medium distance method） 中间距离法与最短距离法和最长距离法一样，也是一种系统聚类法，它定义的类与类之间的距离既不采用两类中样本之间的最短距离，也不采用两类中样本之间的最长距离，而是采用最近和最远之间的距离。

重心法（gravity center method） 重心法为系统聚类法的一种方法。其主要思路是用一个类的重心（该类样本的均值向量）来代表这一类，类与类之间的距离就用重心之间的距离来代表。设 G_p 与 G_q 的重心分别为 $\boldsymbol{x}_p$ 和 $\boldsymbol{x}_q$，G_p 与 G_q 的距离为

$$D_{pq}=d_{x_p x_q}$$

这种定义类与类之间距离的系统聚类法叫重心法。

逐步判别分析法（stepwise discrimination method） 逐步判别分析法是一种有监督的模式识别法，其基本思路与逐步回归基本是相同的，每一步选一个判别能力显著的特征变量进入判别函数，然后，对已选出的特征变量逐个进行检验，检验它们是否对正确判别有显著性意义，如其中有某个变量由于新变量的引入而变为不重要，即对正确判别无显著性意义，则将其剔除，直到判别函数中所包含的所有变量判别能力都显著时为止。

第六节 其他方面的名词术语

一、计算机数字模拟

乘同余法（multiplicative congruential method） 乘同余法是一种用于产生在（0，1）区间均匀分布随机数序列的递推算法，其迭代计算公式为

$$x_i=ax_{i-1}(\mathrm{mod}\ m),r_i=x_i/m$$

其中，x_i 为伪随机数，i 是产生序列号，m 是一个用作模数的大的整数，a 是个整常数。通过调整 a、m 可控制产生的伪随机数系列的特性。上式的意思是：x_i 是 $a\ x_{i-1}$除以 m 后的余数，x_i/m就是第 i 个在（0，1）区间均匀分布的随机数，如此反复迭代，可以产生一个随机数系列 r_1，r_2，…。

刚性微分方程（stiff differential equations） 刚性方程是一类很特殊的微分方程，它类似于线性方程组中的病态体系。很难用一般的微分方程的数值解法来解。目前较有效的方法还推 Gear 法。

混合同余法（mixed congruential method） 混合同余法是一种产生均匀分布随机数系列的数值方法。它迭代计算公式为

$$x_i=a\ x_{i-1}+c(\mathrm{mod}\ m),\quad r_i=x_i/m$$

在此，x_i，i，m 和 a 的意义同前，c 为整常数，一般可取奇数。通过调整 a、c、m 可控制产生的伪随机数系列的特性。

计算机数字模拟法（computer numerical simulation） 计算机数字模拟法是一类利用计算机通过数值算法来模拟化学反应、化学量测过程的研究方法。是研究化学反应、化学量测过程中的误差规律和进行其他化学过程的机理研究的一个很有效的辅助手段，属化学计量学研究的一个重要内容。

均匀分布的伪随机数（quasi-random numbers of uniform distribution） 均匀分布的伪随机数为一种可由计算机产生的均匀分布的随机数系列，要求在指定区间均匀分布，即每个

数有相同的出现概率，数字之间要实际上相互独立，并在指定的长度区间内次序不重复。最常用的两种产生均匀分布的伪随机数的方法是乘同余法和混合同余法。

Monte-Carlo 模拟（Monte Carlo simulation） Monte-Carlo模拟是一种基于统计机理的数字模拟法。二次世界大战期间，美国 Los Alamos 科学实验室的研究人员需要考察核反应的屏蔽问题，他们应用滚轮赌具试验构造理论模式，一步步逼近了问题的解。由于 Los Alamos 的研究项目当时是绝密的，采用著名赌城的名字 Monte Carlo 作为代码，从此，Monte Carlo 模拟方法这一名词便成为随机模拟方法的代名词。

伪随机数（quasi-random numbers） Monte Carlo 模拟的基础是对任意已知分布的数学抽样，即在计算机上产生任意已知分布的随机变量的随机数。最容易得到的是（0，1）区间均匀分布的随机数 r。从均匀分布随机数，通过适当变换即可得到任意分布的随机数。由均匀分布随机数可产生任意分布的随机数。然而，在计算机上产生的随机数，是所谓"伪随机数"，即用数值方法产生的重复再现的（周期性的）"随机"数。取同样的起始值，得到的随机数系列总是全同的。

二、化学构效关系

定量结构活性关系研究（quantitative structure-activity relationship，QSAR） 定量结构活性关系研究是化学计量学的一个十分重要的分支，它通过多元回归或化学模式识别的方法，将化合物的结构信息或结构的量测信息与化学或生物活性联系起来，以找到它们之间的定量关系，用以预测新的有活性的化合物。它也是药物化学中的一个十分活跃分支。

定量结构性质关系研究（quantitative structure-property relationship，QSPR） 定量结构性质关系研究与定量结构活性关系研究相类似，通过多元回归或化学模式识别的方法，将化合物的结构信息或结构的量测信息与化学性质联系起来，以找到它们之间的定量关系。

分子拓扑指数（molecular topological index） 在化学研究中，化合物的结构多用图来表示，通常这样的图称为结构图或分子图。不同的化合物其结构图也不同。如能将不同的化学结构图数量化，并保持分子图中的拓扑性质，这样所产生的指数统称为分子拓扑指数。

分子连接性指数（molecular connectivity index） 分子连接性指数是对 Randic 分支指数加以扩展而得。分子连接性指数有两个基本假设：第一，表示化合物分于结构和它们若干性质间的定量关系所需要的"结构信息"包含在化合物分子的隐氢图中；第二，化合物分子的性质和隐氢图的连接性指数间存在某种函数关系。分子连接性指数一般可由 $^m\chi_t$ 来表示，其中，t 为子图类型，m 为图中连接的边数。

分子识别数（molecular identification number） 分子识别数是一种分子拓扑指数。它实际上是分子连接性指数和分子路径数的一种结合，或更精确的说，是一种加权的分子路径数，其中权值就与分子连接性相关。

局部顶点不变数（local vertex invariants，LOVT's） 局部顶点不变数是分子定量表述中的一个重要概念。巴纳班指出，用实数来表达顶点不变数（vertex invariants）可看成是第三代的分子拓扑指数。第一代分子拓扑指数是以对局部顶点不变数的简单操作而得的整数；第二代分子拓扑指数是对那些由对局部顶点不变数的整数进一步进行复杂操作而得的实数；第三代分子拓扑指数则应是那些由第二代分子拓扑指数进而得到的实数。

立体效应常数（steric effect parameters） 在有机反应性的相关分析中，立体效应是一个很重要的影响因子，所以，提出了不少的反映立体效应常数，其中较著名的有 Taft 立体参

数和 Verloop 的立体参数。

邻接矩阵（adjacency matrix） 化合物的结构多用图来表示，通常这样的图称为结构图或分子图。不同的化合物其结构图也不同，在 1874 年数学家 Sylvester 就指出，一个化学图可用矩阵表示．这个矩阵叫拓扑矩阵或邻接矩阵，此矩阵中的元素 a_{ij}定义为

$$a_{ij}=\begin{cases}0 & \text{原子 } i \text{ 和 } j \text{ 没有键合时} \\ 1 & \text{原子 } i \text{ 和 } j \text{ 没有键合时}\end{cases}$$

这种数字表示法可以把化合物的分子经适当变换存入计算机中。上述矩阵是对称矩阵，矩阵中的每一行或每一列的元素之和即为分子图中相应编号的那个碳原子的点价。

Randic 分支指数（Randic branching index） Randic分支指数是一种在化学计量学研究中应用十分广泛的分子拓扑指数。对于分子的不同异构体，它们在结构式中的主要不同完全是由它们的分支程度不同而定，希望能找到一个“数”，把它们之间在分支程度不同的区别表示出来。Randic 指数正是这样一种简便的“分支指数”它可用来表征分子的分支程度，是分子连接性指数的前身。

退化性或兼并性（degeneracy） 退化性或兼并性是分子拓扑指数研究中的一个十分重要的概念。分子拓扑指数的构建中，经常出现这样一种情况，即不同的分子具有相同的拓扑指数，未能保持分子拓扑指数与化学分子之间的一一对应的关系，这种情况在 QSAR 中被称为退化或兼并。在分子拓扑指数中，分子的 Wiener 数和 Randic 的分支指数都存在着退化或兼并。

Wiener 拓扑指数（Wiener topological index） Wiener最早将拓扑指数引入化学，故以其名命名。其定义为：分子中所有不同碳原子之间的距离总和即为 Wiener 拓扑指数，

$$W(\text{Wiener 数})=(\sum d_{ij})/2$$

后来，此定义被拓广至分子中所有非氢原子之间的距离总和。

有机反应性相关分析（correlation analysis of organic reactivity） 有机反应性相关分析是定量构效关系的前身，是物理有机化学家建立一些物理和化学的参数与有机反应性的相关关系的一种尝试。

三、人工智能与化学专家系统

化学专家系统（chemical expert system） 化学专家系统是由化学家采用人工智能的方法所构建的化学计算机智能系统的总称。所涉及的领域知识当然是化学的专门知识，还涉及建立数据库和知识库、知识表达及推理方式。一般说来，一个化学专家系统至少要包括：一个综合数据库，一组产生式规则和一个控制系统。值得指出的是，当前在化学专家系统的研究中，往往主张将数值方法与基于知识推理的逻辑方法进行柔性集成。

产生式规则表达（generating rule expressing） 产生式规则表达是人工智能用于解决化学课题常用的知识表达方法，这种表达方法包含：① 存储有关问题的状态和性质的叙述性知识的综合数据库，或称事实库；② 存储有关状态转移等规则的过程性知识的规则集或称规则库；③ 用于选择控制策略，将规则与事实进行匹配、控制利用知识推理求解问题的控制器。

最简单的产生式规则有如下格式：

IF（如果）	THEN（则）
前提	结论
条件	行动

产生式系统具有通用性，接近人的自然推理方式，易为用户理解。产生式规则可自由增删、修改，便于用户自行加入领域知识，但简单的产生式系统各规则独立，求解复杂问题时会出现“组合爆炸”，因而效率不高，表达能力亦不够强，需进行扩充，使其前提不限于简单的事实而是事实、情况和条件的综合模式。

广度优先搜索（breadth-first search） 广度优先搜索为人工智能研究中的一种搜索方法。这种搜索方法是从根结点出发，依次在每一水平生成并检验搜索树的各结点，在一个水平的各结点未检验完，即沿广度进行的横向扫描未完成前，不开始生成与检验更深水平的结点（即暂不“扩展”）。这里，认为同一水平各结点对问题的求解是等价的，只是按各结点生成的先后次序，先生成的先检验，沿“广度”遍历所有结点，然后才按原次序将先生成的结点先往后扩展，故称为广度优先搜索法。

逻辑表达（logical expressing） 逻辑表达是人工智能中的一种知识表达方法。知识的逻辑表达通常是指一阶谓词逻辑（FOPL）描述人工智能问题。谓词逻辑表达较严谨精确，具有通用性，接近自然语言，但使用效率低，灵活性较差，常使推理过程冗长。

逆向推理（inverse reasoning） 在人工智能的目标搜索过程中，计算机程序将自动产生各种可能的分支和结点，然后对这些分支和结点进行搜索。直接从起始状态向目标状态进行推理的过程称为正向推理，反之，如果从目标状态出发、来寻找起始状态的过程则称为逆向推理。正向推理亦称是数据推动的，逆向推理又称目标引导的。在用人工智能辅助有机合成路线设计时，就采用逆向推理。

启发式分类（heuristic classification） 启发式知识的应用及启发式搜索是专家系统不同于其他数值计算（包括模式识别）的特征之一。启发式知识是一些不易精确地用数学语言描述的知识，化学专家系统用到化学家的经验，例如谱图解析、结构识别、合成路线设计等均带有假设的色彩。启发式知识可定义为：有关目前问题状况与合适的解之间的经验性知识。而启发式分类是将不同分类层次上相互联系的概念，用不确定联想机制结合起来。启发式分类模型描述了专家系统知识和推理形式的特征，表达了经验性知识的组织和使用。

启发式搜索（heuristic search） 人工智能中的盲目搜索没有用到所涉及问题的领域知识来指导搜索。对于复杂的实际问题，如化学反应的设计常常无法奏效，即由于排列组合太多，出现“组合爆炸”，如果平均每个结点可有 n 个分支，则搜索空间将随深度 d 的增加按 n^d 关系增加。解决的办法是尽可能借助领域知识的帮助来进行启发式搜索，这是人工智能提高求问题解的效率的基本诀窍。实际上，人们在日常生活中就是这样做的。虽然这样得到的解不一定就是获得最优解的最佳途径，但总是在一步步接近目标。启发式搜索的一种方法是局部择优搜索法，即搜索过程中根据领域知识，在一个局部点各可能路径中选择最有希望逼近目标结点的方向，例如爬山选最陡上升方向爬，即梯度函数最大方向搜索，这在只有单峰极值的情况下能奏效。启发式是不确定的、基于典型性假设的，有时是不易理解的一种关系。不确定性是因为启发式认识跳过了大量因果关系的中间层次，而这些中间层次在特定状况下可能不成立，也可能造成错误判断，但正是跳过了中间层次，问题的求解才跨出了一大步。

人工智能（artificial intelligence） 人工智能是一个综合性的科技领域。Hippe 建议将人工智能定义为求知能力的集合，这种能力是指获取知识并用于解决新的（未知的）理论的和/或实际问题能力。人工智能是指借助技术或理论的手段，应用数学方法和/或数理逻辑研究智能的各个方面的问题。简言之，人工智能就是研究如何用人工的方法与技术，即利用计算机等工具，模拟、延伸和扩张人的智能，实现某些机器思维或脑力劳动自动化。人工智能是

计算机和科学技术发展的前沿领域，化学计量学很自然地就成为它的应用领域之一。

深度优先搜索 (depth-first search) 深度优先搜索为人工智能研究中的一种搜索方法。此法进行搜索时，由当前检验的结点作为父结点生成下一级子结点，这样由父结点生成子结点的过程继续下去，看每次检验是否出现目标结点，若未出现，再扩展最晚生成的子结点，如此下去，沿着最晚生成的子结点分支，逐级“纵向”深入发展．故此法称为深度优先搜索法。在到达顶点仍未找到目标结点时，往后“回溯”，在前一个分支处继续沿深度搜索。

搜索树 (searching tree) 用人工智能解决一个问题，常可说作是在各种不同的可能性之间进行选择，或称“搜索”，这个搜索空间可以树分叉的形式表述，常称为搜索树。

与/或图 (AND/OR-graph) 与/或图是人工智能中的一种知识表达方法。与/或图是一种超图，通常以树的形式，亦称与/或树，这种表达知识的方法基于人们求解问题时的两种思维方法。AND 树——用于分解，将复杂的大问题分解成一组简单的小问题，将总问题分解为子问题，若所有的子问题解决了，总问题亦告解决，子问题解决不了的又可继续分解；OR 树——用于变换，将较难的问题变换为较易的等价的问题，后者如能解决，原有难题亦告解决。实际求解问题可兼用二法，即 AND/OR 树表达法。

语义网络表达 (semantic net expressing) 语义网络表达是人工智能中的一种知识表达方法。通过概念及其语义关系用网络图表达知识，用以描述样本、事件、概念等的关系。语义网络起初是为描述人类记忆的心理学模型提出的，在自然语言研究中，表示复杂句型的语义很方便。语义网络中结点表示对象、概念或事件。这种表述法在推理过程涉及较复杂的推理时有用，它较自然、体现了联想思维过程，表达语义关系知识效率较高，但不及逻辑方法严谨，不便于表达判断性知识及动态知识。

知识表达 (knowledge expressing) 人工智能将知识视为高效智能系统的关键要素，知识的表达与管理就成了人工智能研究的中心课题之一。知识可定义为具有智能的人或机器能用以作出合理决策的信息。知识表达是指知识的表示与描述，亦即知识的形式化或模型化。知识有叙述型、过程型与控制型等几种类型。叙述型知识叙述关于系统的状态、环境和条件，问题的构想、定义、事实等；过程型知识表述有关系统状态的变化、问题求解过程的操作等；控制型知识提供如何选择相应的操作、运算和行动的信息。可用于叙述型知识的表达方法有逻辑表达法、语义网络表达法等，而能用于表达过程型知识同时又能表达其他类型知识的表达方法，最典型的是产生式规则表达法。

四、统计学与应用数学

半正定 (positive semidefinite) 半正定为线性代数中所用术语，表示一个矩阵的所有代数余子式的行列式值都大于或等于零，如从正交分解即特征值特征向量的角度看，则是该矩阵的特征值都大于或等于零，一般说来，在化学计量学研究中协方差阵都具有半正定性质。

单位矩阵 (identity matrix) 单位矩阵为线性代数中常用的一种特殊矩阵，即它的对角元素都等于 1，而非对角元素都等于 0。任何矩阵与它的乘积都等于原矩阵。

对称矩阵 (symmetric matrix) 线性代数中，如果一个矩阵它的转置等于其本身，则称为对称矩阵。

欧氏距离 (euclidean distance) 在欧氏空间中，两点 x_i 和 x_j（也可认为是两个矢量）间的直线距离被定义成为欧氏距离，可由 $\|x_i-x_j\|^2$ 表出。在化学模式识别中，欧氏距离一般可用来描述样本在模式空间的相似程度，或度量两个样本在模式空间的远近。

方差分析（analysis of variance） 方差分析为统计学中一种常用方法。它通过对数据的方差进行分解，以得到各部分数据的方差。它在采样理论、回归分析和实验设计中得到了十分广泛的应用。

方阵（square matrix） 方阵是线性代数中的一个概念，表示行列数相等的矩阵。

非奇异矩阵（non-singular matrix） 是线性代数中满秩矩阵的另一种称谓。

广义逆（generalized inverse） 在线性代数中，一般说来，只有满秩方阵才有逆。为了讨论一般矩阵的逆，故提出了广义逆的概念。广义逆有几种，在化学计量学中用得最多的是Penrose-Moore逆，俗称加号逆。

行满秩矩阵（matrix of row full rank） 为线性代数中所用术语，即表示该矩阵的秩与其行的数目一致，也表示该矩阵的行向量之间都是线性无关的。

极大似然估计（maximum likelihood estimator） 极大似然估计为统计学中的一个常用估计方法，最初由Gauss提出，后由统计学家Fisher再次提出极大似然的思想，使之得到广泛研究和应用。似然估计方法的基本思路是通过一族含未知参数的密度函数来构建一个似然函数，继求得该量学中也得到了广泛的应用。

矩阵的迹（trace of matrix） 一个方阵的对角元素的和称为该矩阵的迹，是线性代数中的一个十分重要的概念。由于正交变换时矩阵的迹不发生变化，所以它在因子分析的误差理论中起着十分重要的作用。

矩阵（matrix） 在分析化学量测中，一个化学样本一般都可由一个谱来表示，如常见的光谱、色谱、各类波谱和各种电化学谱。如果将很多样本收集在一块，就可得到一个矩阵。一般说来，在这样的数据矩阵中，对一个样本量测所得到的一个谱可置为一个行矢量，这样的行矢量可称为一个样本对象或一个样本；与之类似，在这样的数据矩阵中，一个变量（如一个分析通道，一个波长等）可置为一个列矢量，这样的列矢量可称为一个变量。大多数化学计量学方法，都是建立在这样的化学量测矩阵之上。

可靠性分析（reliability analysis） 可靠性分析是统计学中质量控制研究中的一个重要分支，在化学计量学的研究中也占有十分重要的地位。对化学计量学方法的结果进行可靠性分析是十分必要的一个步骤。

幂等矩阵（idempotent matrix） 幂等矩阵是线性代数中一个重要概念。它是指一类自身相乘还得到自身的矩阵，即 $\boldsymbol{AA}=\boldsymbol{A}$ 的矩阵。

Moore-Penrose 逆（Moore-Penrose inverse） Moore-Penrose逆又称“+”逆，记为 $\boldsymbol{A}^{+}$。对一个 $n\times m$ 阶矩阵 $\boldsymbol{A}$，若存在一个 $m\times n$ 阶矩阵 $\boldsymbol{B}$，要求满足以下四个条件：

① $\boldsymbol{AA}^{+}\boldsymbol{A}=\boldsymbol{A}$； ② $\boldsymbol{A}^{+}\boldsymbol{AA}^{+}=\boldsymbol{A}^{+}$；

③ $(\boldsymbol{AA}^{+})'=\boldsymbol{AA}^{+}$； ④ $(\boldsymbol{A}^{+}\boldsymbol{A})'=\boldsymbol{A}^{+}\boldsymbol{A}$

条件①和条件②称为反身性；条件③和条件④称为对称性。容易证明，Moore-Penrose逆是存在且唯一的。Moore-Penrose逆在化学计量学中应用较广。

内积（inner product） 内积又称点积，是矢量运算中的一个十分重要的运算规则，所得的是一个数。它的数学定义为

$$\boldsymbol{x}'\boldsymbol{y}=\sum x_i y_i$$

它的几何意义为两矢量间的投影。如果两个矢量的内积为零，则说明该两矢量垂直。

奇异矩阵（singular matrix） 奇异矩阵是亏秩矩阵的另一种称谓。

奇异值分解法（singular value decomposition，SVD） 奇异值分解法是计算数学中对数

据阵进行正交分解的一种计算方法，任意实数矩阵经奇异值分解后，一般可得到三个矩阵，即一个列正交矩阵，一个对角矩阵，一个行正交矩阵。其中列正交的矩阵在化学计量学中常称为得分矩阵，行正交的矩阵在化学计量学中常称为荷载矩阵，对角矩阵则收集了原矩阵的特征值。这一方法无论在多元校正与多元分辨中还是在化学模式识别中都得到了十分广泛的应用。

双线性矩阵（bilinear matrix） 双线性矩阵在化学计量学中表示一个矩阵是由两个矢量的外积而得。这样的矩阵的秩为1。联用色谱和二维荧光光谱仪所产生的纯物质矩阵都是双线性矩阵。

特征值和特征矢量（eigenvalue and eigenvector） 特征值和特征矢量是线性代数中的一个十分重要的概念，在化学计量学和量子化学中得到了非常广泛的应用。它们的基本数学定义为：一个矩阵 A，如果有

$$A\gamma_1=\lambda_1\gamma_1,\cdots,A\gamma_k=\lambda_k\gamma_k$$

则称满足上述方程的矢量 $\gamma_i(i=1,2,\cdots,k)$为矩阵 A 的相对于特征值 $\alpha_i(i=1,2,\cdots,k)$的特征矢量。一般说来，在化学计量学中，特征值所代表的是数据阵的方差，特征矢量是张成数据阵所代表的线性空间的基向量。

投影寻踪法（projection pursuit） 投影寻踪是一种近年来在统计学中得到迅速发展的多元统计的方法。如果一个化学量测的 m 维数据可看成是 m 维空间的一个点，则一组这样的化学量测数据可看成是 m 维空间的一簇点云。降维处理技术的目的就找到一种方法，将这些高维数据点在低维空间中描述出来而尽量做到不改变这些高维点云的基本结构。投影寻踪法（PP）就是这样一种降维技术。它首先定义一种投影指标，然后，找到一种最优方法来实现这种投影。不同的投影指标就定义了不同的投影寻踪方法。

投影阵（projection matrix） 投影阵是线性代数中一种十分重要的矩阵，其基本特征为对称幂等。在化学计量学中，投影阵得到了非常广泛的应用，很多多元校正和多元分辨的化学计量学算法都是基于投影阵而提出的。

样本对象（object） 在化学计量学中，人们常把一个化合物的量测谱（多元校正与多元分辨）或是描述一个样本的一组参数（化学模式识别）称为一个样本对象，在量测矩阵中它通常为一个行向量。

中位数（median） 中位数是统计学中的一个十分重要的参数，它与平均值相类似，但是平均值不稳健，而中位数则是一个十分稳健的指标。

正交投影阵（orthogonal projection matrix） 正交投影阵是线性代数中的一种特殊矩阵，它们都是对称幂等矩阵，其几何意义是将某一矢量和矩阵向其正交补空间投影，在化学计量学，特别是在多元校正和多元分辨中得到了十分广泛的应用。

正态分布（normal distributions） 设连续型随机变量 x 的概率密度为

$$f(x)=(1/\sqrt{2\pi}\sigma)e^{-(x-\mu)^2/(2\sigma^2)} \qquad -\infty\leqslant x\leqslant\infty$$

其中，μ，$\sigma\geqslant0$ 为常数，则称 x 服从参数为 μ，σ 的正态分布或高斯分布，记为 $x\sim N(\mu,\sigma^2)$。正态分布是一种十分重要的一种分布，一般说来，正态分布表示由多种不定因素而产生出来一种分布，所以，大都假设化学量测误差服从正态分布。用以检测一系列变量是否服从正态分布的最简易方法是采用正态分布图来对它们进行检验。

正定（positive definite） 正定是线性代数中一个很重要的概念，如果说一个矩阵是正定的，则说明这个矩阵的特征值都大于零，如果说一个矩阵是半正定的，则说明这个矩阵的特征值都大于等于零。一般说来，协方差矩阵都是正定或半正定的。

第三章　分析采样理论和方法

采样是分析测试工作的第一步，分析测试结果的可靠性与采样是否正确直接相关。分析测试的目的就是要根据从局部试样（样本）测得的数据来获取有关对象全体（总体）的无偏信息。怎样使局部采样可在统计意义上尽可能地代表总体，是采样理论和方法所研究的内容。本章将分别讨论分析化学中采样理论和方法。采样理论是指如何进行试样采集的数学统计理论。本章对常用的采样方法亦分别加以介绍，如固体物质的采样方法、动态过程的采样方法和质量检验的采样方法。介绍理论将以其基本思想为主线，不作任何推导[1]；介绍基本方法则以其基本思路为主线，指出其使用特点和适用范畴。

一般说来，一个成功采集的分析试样（或称样本），从统计上应满足下述要求：① 样本均值应能提供总体均值的无偏估计，一般而言，随机采样是保证这种无偏性的基本方法；② 样本分析结果应能提供总体方差的无偏估计，例如系统试样，应能提供分析对象有关参量随时间的变化等；③ 在给定的时间和人力消耗下，采样方法应给出尽可能精密的上述估计。

第一节　采样的基本概念和理论

一、随机采样

随机采样系指等概率地从总体中采集的试样，采样应在随机状态下进行，例如将分析对象全体划分成不同编号的部分，再根据随机数表进行采样，这种采样法亦称概率采样。在分析实践中，要区分目标总体和母总体两个概念。目标总体是指欲根据采样与分析作出相应结论的目标对象；而母总体则是实际被采集试样的对象，这两者很少一致，但我们希望其区别尽可能小。采集随机试样的方法即随机采样就是尽可能缩小这一差别的一种手段。

对于随机采样，如果每 n_s 样本被分析了 n_a 次，则其总方差 σ_0 为

$$\sigma_0^2=\sigma_s^2/n_s+\sigma_a^2/(n_s n_a) \tag{3-1}$$

在此，σ_s^2 和 σ_a^2 分别表示采样和分析方差。式(3-1)可用于随机采样设计。假设

$$\sigma_a^2=\alpha\sigma_s^2$$

则式(3-1)可写成

$$\sigma_0^2=\sigma_s^2/n_s+(\alpha/n_a)(\sigma_s^2/n_s)$$

从此式可以得出下述结论：

① 对于给定的 α、n_s 和 n_a，总方差是随着采样方差增加而增加。

② 对于给定的总分析次数（$n_s n_a$），如果不考虑分析成本，则随机采样应尽可能保证采样次数多为好。如对 6 个随机样本进行两次分析要比对 4 个随机样本进行 3 次分析的总方差要小。

③ 随机采样的总方差是 α 的线性函数。当 α 为一很小数，即分析测定的方差比采样方差小得多时（在实际中通常是这种情况），(α/n_a) (σ_s^2/n_s) 比起 (σ_s^2/n_s) 来就可以忽略。对于这样的情况，Youden 曾指出[2]，当分析误差下降到采样误差的 1/3 或更低时，宁可使用快速简便的、精密度不高但能与采样误差匹配的方法进行分析。其理论根据就在于此。

实际上，式（3-1）同样可用于结合分析成本一起计算的情况。设采样成本和分析一次试样的成本分别为 C_s 和 C_a，则总成本 C 为

$$C=n_sC_s+n_sn_aC_a$$

考虑到式（3-1），总成本可表示为

$$C=[\sigma_s^2/\sigma_0^2+\sigma_a^2/(\sigma_0^2n_a)](C_s+n_aC_a)$$

对此式就 n_a 进行微分并令其等于零，可得到在固定方差条件下使总成本最小时最佳的样本分析次数

$$n_a=(\sigma_a/\sigma_s)(C_s/C_a)^{1/2}$$

和采样的数目

$$n_s=[\sigma_s^2+(\sigma_a^2/n_a)]/\sigma_i^2$$

二、系统采样

系统采样系指为了检验某些系统假设而采集的试样，例如生产或其他过程中成分随时间、温度的变化而在空间中变化，这种场合下的采样问题有重要的实际意义[3]。一般是间隔一定区间（时间、空间、区域）采样，间隔不一定是等距的，有时，事先可预期总体成分是不均匀的，系统采样要尽量减少这种不均匀性的影响。对于这样的情况，可采用分层采样。系统采样的误差分析与随机采样是相似的。

三、分层采样

当分析对象可划分为若干采样单元时，随机采样可从总体的全体采样，亦可分层或分步采样。当被划分的各采样单元之间试样成分的变化显著大于每一单元内部成分变化的情况时，分层采样是最好的选择。分层采样是先将分析对象划分成不同的部分或层，然后对不同的层次进行随机采样。此时，总方差为

$$\sigma_0^2=\sigma_b^2/n_b+\sigma_s^2/(n_bn_s)+\sigma_a^2/(n_bn_sn_a) \tag{3-2}$$

式中 n_b——采样层数；

σ_b^2——层之间的方差；

n_s——每层的采样次数；

σ_s^2——层内的方差。

当分析对象是总体均匀分布的话，分层采样就与随机采样是一样的，但如果层间方差与层内方差显著不同时，分层采样将明显优于随机采样。值得指出的是，由式（3-2）表出的方程不可能分别同时唯一地求出 n_b、n_s 和 n_a，有必要在它们之间进行适当调整和妥协。如考虑进行分层的成本为 C_b，对每层的采样成本为 C_s，分析试样的成本为 C_a，则整个分层采样过程的总成本为

$$C=n_bC_b+n_bn_sC_s+n_bn_sn_aC_a$$

当方差固定时，在使得总成本最小的目标下，n_b、n_s 和 n_a 的最优选择可由式（3-3）的 3 个式子给出，即

$$n_b=\sigma_b(\sigma_bC_b^{1/2}+\sigma_sC_s^{1/2}+\sigma_aC_a^{1/2})/(\sigma_0C_b^{1/2}) \tag{3-3a}$$

$$n_s=(\sigma_s/\sigma_b)(C_b/C_s)^{1/2} \tag{3-3b}$$

$$n_a=(\sigma_a/\sigma_s)(C_s/C_a)^{1/2} \tag{3-3c}$$

四、代表性采样

代表性采样一般是指特定的分析项目所涉及的采样，例如按环境保护部门规定采集废水试样就是这种例子。在分析化学的实际工作中，代表性采样是一种分层采样的特殊情况，这种情况的分层采样可对目标成分提供总体均值的无偏估计。对于在分层采样中每层的大小和方差均不相同的情况下，为了得到总体均值在方差最小条件下的无偏估计，在第 k 层的采样数目 n_{sk}，应与该层的大小 w_k 和标准差 $(\sigma_s)_k$ 有关，即

$$n_{sk}/n = [w_k(\sigma_s)_k] \Big/ \sum_k^n [w_k(\sigma_s)_k] \tag{3-4}$$

如果在每一层的标准差都相等，则式(3-4)可变为

$$n_{sk}/n = w_k \Big/ \sum_k^n w_k \tag{3-5}$$

式（3-5）说明，每层的采样数是与该层的大小成正比的。同时，还说明这样的采样是与随机采样不同的。很多的分析技术规则都给出了怎样进行代表性采样的规定。代表性样本是由权威性组织为某种特殊目的而制成的样本。一般说来，在制作代表性样本的过程中，主要须考虑的就是上述讨论的式（3-4）及式（3-5）两个式子。显然，总体均值 x 的无偏估计应该是各分层均值的加权均值，即

$$\bar{x} = \frac{w_k}{\sum_k^n (w_k x_k)} \tag{3-6}$$

复合试样也是制取代表性试样的一种方式，将一些采集的单个试样混合起来作为复合试样，必须考虑这样做能否取得正确的有代表性的结果。

分层采样的方差分析 式（3-2）中的方差如 σ_b^2、σ_s^2 和 σ_a^2 都是未知的，但是，如果对所有采样样本都进行了分析以后，通过方差分析的方法就可将它们都分别求出。设 x_{kji}表示第 k 层的第 j 个样本的第 i 次的某一目标化学成分的测定值，$\bar{x}_{kj}$表示第 k 层的第 j 个样本的测定均值，对于整个分层采样的过程，有 $n_b n_s n_a$ 个样本，则 σ_a^2 的估计可由下式给出

$$V_a = \sum_{k=1}^{n_b}\sum_{j=1}^{n_s}\sum_{i=1}^{n_a}(x_{kji} - \bar{x}_{kj})^2/[n_b n_s(n_a - 1)] \tag{3-7}$$

对于第 k 层，可以得到该层的采样方差估计

$$V_s = \sum_{k=1}^{n_b}\sum_{j=1}^{n_s}(\bar{x}_{kj} - \bar{x}_k)^2/[n_b(n_s - 1)] \tag{3-8}$$

在此，$\bar{x}_k$ 为第 k 层的样本均值。这样计算的已不是单纯总体采样方差 σ_s^2 的估计，它还包含了 V_a 的贡献。如设采样误差与分析测试误差相互独立，则 V_s 是 $\sigma_s^2+(\sigma_a^2/n_a)$估计值。对于层间方差，可由下式求出

$$V_b = \sum_{k=1}^{n_b}(\bar{x}_k - \bar{x})^2/(n_b - 1) \tag{3-9}$$

在此，$\bar{x}$为总体的样本均值。

同理，这样计算的层间方差并不就是由式(3-2)中的 σ_b^2，它还包含了层内采样和分析测试的方差的贡献，是$[\sigma_b^2+(\sigma_s^2/n_s)+(\sigma_a^2/n_s n_a)]$的估计值。但只要利用式(3-7)～(3-9)计算出了 V_a、V_s 和 V_b，则不难通过它们将 σ_b^2、σ_s^2 和 σ_a^2 估计出来。实际上，对于所有的 $n_b n_s n_a$ 样本的方差，有如下的方差分解式：

$$\sum_{k=1}^{n_b}\sum_{j=1}^{n_s}\sum_{i=1}^{n_a}x_{kji}^2 = n_b n_s n_a x^2 + n_s n_a \sum_{k=1}^{n_b}(x_k - x)^2$$
$$+ n_a \sum_{k=1}^{n_b}\sum_{j=1}^{n_s}(x_{kj} - x_k)^2 + \sum_{k=1}^{n_b}\sum_{j=1}^{n_s}\sum_{i=1}^{n_a}(x_{kji} - x_{kj})^2$$

方差分析本质上就是建立在此分解式的基础之上。表 3-1 列出了分层采样的方差分析表，利用此表可以方便地进行分层采样的各种方差的估计。

表 3-1　分层采样的方差分析表

变化来源	平方和	自由度	均方值	均方值的期望
均　值	$n_b n_s n_a x^2$	1		
层　间	$n_s n_a \sum_{k=1}^{n_b}(x_k - x)^2$	$n_b - 1$	$n_s n_a V_b$	$\sigma_b^2 - (\sigma_s^2/n_s) - (\sigma_a^2/n_s n_a)$
采　样	$n_a \sum_{k=1}^{n_b}\sum_{j=1}^{n_s}(x_{kj} - x_k)^2$	$n_b(n_s - 1)$	$n_a V_s$	$\sigma_s^2 - (\sigma_a^2/n_a)$
分　析	$\sum_{k=1}^{n_b}\sum_{j=1}^{n_s}\sum_{i=1}^{n_a}(x_{kji} - x_{kj})^2$	$n_b n_s(n_a - 1)$	V_a	σ_a^2
总　计	$\sum_{k=1}^{n_b}\sum_{j=1}^{n_s}\sum_{i=1}^{n_a}x_{kji}^2$	$n_b n_s n_a$		

五、最小采样数目的估计

前面所谈到的有关最佳采样数目的估计都是建立在真实采样方差和真实分析方差之上的。但实际上采样乃是建立在相对小样本采样之上，所以怎样利用小样本采样的各种方差所得的估计值来进行最小采样数目的估计是要讨论的问题。本小节所要讨论的最小采样数目的估计方法是建立在学生分布统计量或称 t-统计最的基础之上的[4]。

根据学生分布，可通过计算所得的均值$\bar{x}$来对真实均值 μ 作出如下的区间估计

$$\mu = \bar{x} \pm (ts_0)/\sqrt{n} \tag{3-10}$$

式中，s_0 是总标准偏差 σ_0 的估计；学生分布参数 t 为取一定置信度和自由度$(n-1)$时的对应值。据此可以算出 n

$$n = (ts_0)^2/(\bar{x}-\mu)^2 = (ts_0)^2/e_0^2 \tag{3-11}$$

因 n 为一待求数，所以在对参数 t 查表取值时先用 $n=\infty$作为其自由度来确定 t 值，用此 t 值根据(3-11)式算出一个 n 后，继用此 n 来再查得一个新的 t 值，如此循环，直到 n 收敛于一常数。对于随机采样的情况，总的分析测试数目为 $n_s n_a$，如果分析测试的标准偏差 s_a 很小可以忽略时，则分析测试的误差也可以忽略，因此可以将(3-11)式简写为

$$n_s = (ts_s)^2/e_s^2 \tag{3-12}$$

式(3-12)是由小样本采样导出的。对于大量样本采样的情况，即采样量占总体量具有相当部分时，需引入一个称为"有限总体校正"因子，即$(1-n/N)^{1/2}$，来进行校正。在此，N 是总体所含样本数。将此校正因子乘入式(3-10)右边的第二项，得

$$n_s = (t^2 s_s^2 N)/(e_s^2 N + t^2 s_0^2) \tag{3-13}$$

从式(3-13)明显可知，有限总体校正因子只对小总体的情况是适用的，如果总体很大，也即 N 很大，则该校正因于就变为 1 了。

引入 t-统计量和置信度的概念，就使得对采样数目的估计变得实用了。这样，在前面所讨

论一些式子的未知真实量如 σ 就可用估计方差来替代了，如式(3-1)就可变为

$$e_0^2=(t^2s_s^2)/n_s+(t^2s_a^2)/(n_sn_a) \quad (3\text{-}14)$$

式(3-14)中的 t 就是上面所讨论的 t-统计量。同理，可据此求得随机采样的样本数 n_s

$$n_s=[t^2(s_s^2+s_a^2/n_a)]/e_0^2 \quad (3\text{-}15)$$

六、采 样 常 数

（一）Ingamell 采样常数

为表征混合得很好的实验室样本的均匀性，Ingamell 定义了一个采样常数 K_s

$$K_s=R^2m \quad (3\text{-}16)$$

式中，R 为相对标准偏差，即 $R=100\times s_s/\bar{x}$，m 为被分析样本的质量。如将式（3-16）与式（3-12）比较一下可发现，式（3-16）中隐含如下意义：Ingamell 采样常数 K_s 相当于保证采样相对标准偏差为 1%时的必需样本质量，此时，t-统计量的取值为 1，即自由度为无穷大和置信度为 68%时的 t 值。对于 1g 样本，其 Ingamell 采样常数就是经一个很精确的分析方法（因此时分析测定的误差相对于采样误差可以忽略）测定所得的相对标准偏差的平方根，一般就可用此方法来实验确定 Ingamell 采样常数。如果将式（3-16）改写，可得

$$R^2=K_s/m \quad (3\text{-}17)$$

因 K_s 为一常数，所以，采样的相对偏差与采样质量呈反比，换言之，欲得到很低的采样的相对偏差就必须保证采样需有相应的足够质量。同时，从式(3-17)还可以看出，K_s 愈小，说明样本的混合程度愈好。对于分层测定的非均匀样本，其不同的层次当有不同的 Ingamell 采样常数，人们也常用此法来测定总体的均匀程度。对于分隔的总体，应引入附加的分隔常数。

（二）Visman 采样常数

Visman 阐述了在考虑分隔效应下的采样理论[4]。实验估计方差与两个采样常数有关，一个是与 Ingamell 采样常数相似的均匀度常数 A，另一个是反映分隔（segregation）程度的常数 B。

$$s_s^2=A/(mn_s)+B/n_s \quad (3\text{-}18)$$

式中，mn_s 是 n_s 个样本的总质量。如果总体样本是均匀的，则 $B=0$，此时式（3-18）就与式（3-16）完全类似，并可容易发现 Ingamell 采样常数与 Visman 采样常数存在如下关系：

$$A=10^{-4}\times\bar{x}^2\times K_s$$

Visman 采样常数可通过收集一系列小样本和一系列大样本来进行实验确定。先对这两系列的方差进行估计。然后，将这两系列样本的质量和所得的方差代入式（3-18），就可估计出 Visman 采样常数 A 和 B 来。同时，从式（3-18）还可以看出，假设希望保持方差不变，采样数是随着 Visman 的表征分隔效应的常数 B 增加而增加的，而且，采样方差与采样数成反比，增加采样数可降低采样方差。当 $B=0$ 时，说明样本总体是均匀的，此时，只要样本的总质量（mn_s）不变，采样数将不再影响采样方差。

第二节　非均匀体系建模方法与大批物质的采样误差

一、固体物质的采样理论和方法

Benedetti-Pichler 将大批固体物质的采样问题与在一个装有成百吨混合的白豆和红豆大仓库中对它们进行计数的问题作了一个类比。如采要对仓库中所有的豆子都一一计数，则需花极长的时间，人们只好转向一个比较现实但不予全部计数精确的方法。首先，从中取出一

个样本分别对白豆和红豆计数，这样计数的精确度显然是与所取样本的大小有关。从统计上讲，这种以局部代表全体的办法将带来误差，取样愈少，误差愈大。可通过二项式分布来决定采样量。对于大批固体物质总体的现代采样理论一般说来都是基于这一道理。

设分析对象总体是由两类立方体颗粒 A 和 B 所组成，这些颗粒具有同样的棱长 μ 和同样的密度 ρ，颗粒 A 的总质量百分比为 $w_A(\%)$，而颗粒 B 的总质量百分比为 $w_B(\%)$。先考虑一个最简单模型，即只设颗粒 A 中含有待测物 x，且此待测物在颗粒 A 的质量百分比为 ${}^Aw_x$。如取具有质量为 m 的样本进行化学分析，其采样方差为 σ_s^2。这样的模型可用二项式分布来处理。对于服从二项式分布的随机变量，其方差为 npq。根据上述讨论的简单模型，采样颗粒数 (n) 应为 $m/\rho\mu^3$，其取得颗粒 A 的概率 (p) 应为 $w_A(\%)$，其取得颗粒 B 的概率 (q) 应为 $w_B(\%)$ 或 $(1-w_A)$，据此，可得根据颗粒数来表述的采样方差 σ_s^2

$$\sigma^2_{s(\text{particle})}=[m\times w_A\times(100-w_A)]/[(\rho\mu^3)\times1000] \tag{3-19}$$

如以所求物质的百分比来计算 $\sigma_{s(\text{particle})}$，则应再除以颗粒的数目，即

$$\sigma_s=[\sigma_{s(\text{particle})}/(w/\rho\mu^3)]\times100$$

最简单的情况是颗粒 A 就是所求的纯物质，即此时有 $w_A={}^Aw_x$，那么

$$\sigma_s=(\rho\mu^3/m)\times{}^Aw_x\times(100-{}^Aw_x) \tag{3-20}$$

式(3-20)可用于估计给定样本质量时的采样误差，或用来在给定了采样误差标准的条件下来估计所需最小采样量。当然，此模型过于简单，在实际的分析化学实践中，一般须对此模型进行必要的修订。

上述统计原理，可推广到较接近实际的情况，即设试样总体中含两种颗粒 A 和 B，颗粒 A 含待测物的百分比（质量分数）为 ${}^Aw_x$，颗粒 B 含待测物的百分比（质量分数）为 ${}^Bw_x$，且颗粒 A 的密度为 ρ_A，颗粒 B 的密度为 ρ_B，这样，上述简单模型可变为

$$\sigma_s^2=[(\rho_A\rho_B/\rho)\times(\mu^3/m)\times w_A\times(100-w_A)]\times[({}^Aw_x-{}^Bw_x)^2/100] \tag{3-21}$$

式(3-21)中，不但考虑了试样总体的密度 ρ，而且，对于颗粒 A 和颗粒 B 的不同密度也一起进行了考虑，同时此式中的最右边的平方项，即$[({}^Aw_x-{}^Bw_x)^2/1000]$还将两者颗粒的不同百分含量也进行了考虑。从式(3-21)可明显看出，${}^Aw_x$ 和 ${}^Bw_x$ 的差别越大，其采样误差也越大。

在分析化学文献中[5~7]，对上述模型有很多不同的表达式，然而它们之间的差别只在于一些微小的统计近似和假设的不同，本质上是类似的，下面将它们列出，以备读者参考。

① Benedetti-Pichler 经典方程

$$\sigma_s^2=(\rho_A\rho_B^2/\rho^2)\times[p(1-p)/n]^{1/2}\times[({}^Aw_x-{}^Bw_x)/w_x]^2 \tag{3-22}$$

式中，w_x 为待测组分的总体百分含量(质量分数)，p 为取得颗粒 A 的概率。

② Ingamell 方程

$$K_s=R^2w=\frac{({}^Bw_x\rho_B-{}^Aw_x\rho_A)^2({}^Bw_x-w_x)(w_x-{}^Aw_x)\mu^3}{[\rho_B({}^Bw_x-w_x)({}^Bw_x-{}^Aw_x)+\rho_A(w_x-{}^Aw_x)({}^Bw_x-{}^Aw_x)]\times10^{-8}/x^2} \tag{3-23}$$

③ Wilson 方程

$$\sigma_s^2=\frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k}[(\Delta w_i\rho_i-\Delta w_j\rho_j)/100]^2\times(m_im_j/\rho_i\rho_j)\times(\mu^3\rho/m) \tag{3-24}$$

Wilson 方程是将二元颗粒的非均匀体系推广到多元颗粒的非均匀体系的表达式。式中下标 i 和 j 表示在 k 种颗粒的第 i 或第 j 种颗粒，Δw_i 表示 w_i 与总体含量 w_x 之间的差值。同时，Wilson 方程是建立在所有颗粒都具有相同体积的基础上，如果颗粒的密度具有较大的不同，

Wilson 方程只在 s_s 不太大时才成立。

二、颗粒性质因子（Gy 理论）

基于研究粒状物质的采样，Gy 对非均匀体系的采样发展了一套较全面的理论。根据 Gy 的采样理论，由式（3-20）表出的简单公式可改写成

$$\sigma_s^2=(\rho\mu^3)\times w_x\times(100-w_x)\left(\frac{1}{m}-\frac{1}{M}\right)fgl \qquad (3\text{-}25)$$

在式(3-25)中，原式中的$(1/m)$由$\left(\frac{1}{m}-\frac{1}{M}\right)$代替了。这样的替换主要反映了总体物质的质量对采样方差的影响，采样误差是随着采样质量 m 的增加和总体质量的减少而减少的，当采样质量 m 增加到接近于总体质量 M 时，采样方差就会接近于零；当总体质量 M 比采样质量 m 大很多时，$\left(\frac{1}{m}-\frac{1}{M}\right)$就很接近$(1/m)$，总体质量的影响可忽略不计了。

1. 颗粒形态因子(f)

在式(3-20)的导出过程中，所有颗粒都是假设为具有立方体形体积(μ^3)，可实际上颗粒的形态是有差别的。式(3-25)中引入的 f 因子就为了补偿这些差别的。如果是在理想的情况下，颗粒形态因子 f 可以取 1；如果所有颗粒都是以 μ 为直径的小圆球组成时，则颗粒形态因子 f 可取 0.524。对于大部分总体样本，f 可取 0.5，相对于 f 取 1，这是一种对实际情况的较好的逼近。如果总体物质是具有特殊形态的颗粒，如黄金颗粒，颗粒形态因子 f 可以低至 0.2。

2. 颗粒大小分布因子(g)

在式(3-20)的导出过程中，所有颗粒都是假设为具有相同大小的，可实际上颗粒的大小也是有差别的，故在式(3-25)中引入颗粒大小分布 g 因子来补偿这种差别。在理想的情况下，颗粒大小分布因子 g 取 1；如果颗粒大小差别较大，即既有很粗的又有很细的混在一起时，则颗粒大小分布因子 g 推荐取 0.25。对于只有中等颗粒或是只有小颗粒时，则颗粒大小分布因子 g 可分别取 0.5 或 0.75 为好。

3. 释放(Liberation)因子(l)

在式(3-20)的导出过程中，假设总体物质由两种颗粒 A 和 B 所组成，而且，还假设了 ${}^{A}w_x=w_x$。可是，在实际情况下，有可能一些板块可能同时含有两种颗粒。在研磨过程中，那些含有待测物的颗粒从板块中被释放出来。如将这种释放过程也加以考虑，就可引入释放因子 l，其定义为

$$l=(\mu_1/\mu_2)^{1/2} \qquad (3\text{-}26)$$

在此，μ_1 是板块中含有待测物的细粒的平均尺寸，而 μ_2 则是总体物质中的最大板块颗粒的尺寸。当 $\mu_2\leqslant\mu_1$ 时，$l=1$，即此时没有释放效应。释放因子 l 的估计可由显微镜或筛选实验来确定。

4. 组成因子(c)

除$\left(\frac{1}{m}-\frac{1}{M}\right)$、$f$、$g$ 和 l 外，还可以定义一个组成因子。这个因子主要反映颗粒组成变化对采样方差的贡献。考虑到总体物质的密度(ρ)实际应为两种不同颗粒密度(ρ_A 和 ρ_B)的加权平均，即

$$\rho=\frac{x}{100}\rho_A+\frac{100-x}{100}\rho_B \qquad (3\text{-}27)$$

则组成因子 c 可定义为

$$c=x(100-x)\left(\frac{x}{100}\rho_{A}+\frac{100-x}{100}\rho_{B}\right)$$

此时，Gy 的采样理论可由一个很简单的式子表示

$$\sigma_s^2=\mu^3\left(\frac{1}{m}-\frac{1}{M}\right)cfgl \tag{3-28}$$

三、动态过程中物质流的采样理论和方法

若采样对象是一个连续过程，需从过程中的物质流中采样，则情况较为复杂。从数学上讲这种过程可以时间序列表示。在这个序列中，可在图 3-1 中取长度 p 作为一个“批量”，按频率 A 在时间 G 内采样 n 次，采样理论需考察 G、A、n 对估计 p 的成分的影响[8,9]。对于一阶 Gauss 稳态随机过程，令 $p=nA$，设 $A>G$，即采样不重叠，则采集的总样 $S=nG$。这里，采样的总样成分变化的方差 σ_m^2，与作为批的整个 p 段的成分变化方差 σ_μ^2 对采样方差 σ_s^2 的贡献可表述为

$$\sigma_s^2=\sigma_m^2+\sigma_\mu^2+2\sigma_{m\mu} \tag{3-29}$$

式中，$\sigma_{m\mu}$为总样与批二者的协方差。

Muskens 等[10~12]从一阶 Gauss 稳态随机过程理论推导了式（3-29）右边的各项。设过程的相关因子为 T_x，记 P/T_x 为 p，G/T_x 为 g，A/T_x 为 a，过程方差为 σ_x^2，则

$$\sigma_m^2=(2\sigma_x^2/ng)\{g-1+e^{-g}+(e^{-g}+e^{g}-2)[e^{-a}/(1-e^{-a})]-[e^{-a}(1-e^{-p})]/[n(1-e^{-a})^2]\} \tag{3-30a}$$

$$\sigma_\mu^2=(2\sigma_x^2/p^2)(p-1+e^{-p}) \tag{3-30b}$$

$$\sigma_{m\mu}=(\sigma_x^2/npg)\{2ng+(1-e^{-p})[(e^{-g}-1)/(1-e^{-a})+(e^{g}-1)/(1-e^{-a})]\} \tag{3-30c}$$

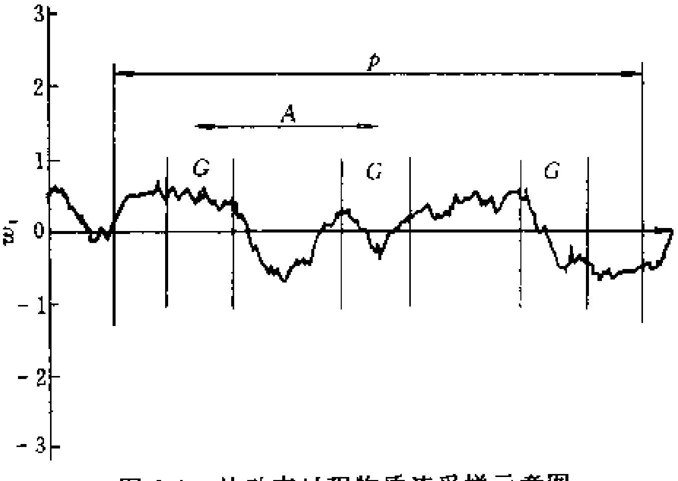


图 3-1　从动态过程物质流采样示意图

这里，作为批的参量起作用的是 p（以 T_x 单位表征 p）。批的概念在这里并不限于一个均相的物质流，例如管道中液体物质对应于 p 的量可以是时间、长度、质量等任何单位表示，以时间表示时，相关因子 T_x 称为时间常数。亦可使用无因次，或以物质的包数等表示。在上述式子中，总样的参量包括 G 的大小及两次采样之间的距离 A，如果选择 G 很大，以致采样之间实际无间隔，显然此时 $\sigma_s^2=0$，这时整个“批”均被采集，$A=p/n$，总样的大小是 $nG=p$。随时间常数增大，即 $T_x\to\infty$，体系趋近于一均匀的对象，这时 σ_m、σ_μ、$\sigma_{m\mu}$均趋近于零，取一份任意大小的样即能代表总体[9]。

第三节　质量检验的采样方法

质量检验分析中，需根据分析结果决定产品是否合格，一般分为计量抽样检验与计数抽样检验两种采样检验方法[6]。

一、计量抽样检验

这种检验方法是定量测试产品中某一变量。分析检验中常见的情况是测定某一组分的百分含量，例如某种有效成分的最低含量，某种有害成分的最高含量等。在考虑这种检验的采样时，首先需确定合格产品的相应定量标准。今假设待测的组分含量这一变量服从正态分布，并设产品中的有效成分 x 的最低含量标准为 x_0，分析方法的标准差为 σ，可计算对应与任意真实含量的 x 的 u 值，即标准正态分布值：

$$u=(x_0-x)/\sigma \tag{3-31}$$

从标准正态分布可计算实际含量为 x 时该批试样被接受的概率 p。例如，设某工厂购买氟硅酸原料，要求 H_2SiF_6 的含量不得低于 350g·L^{-1}，分析方法单次测定的标准差为 10g·L^{-1}，根据 u 值可得表 3-2 的数据。

表 3-2　不同实际含量 x 的试样被接受的概率

x/(g·L^{-1})	u	p	x/(g·L^{-1})	u	p
310	4	0.00003	350	0	0.5000
320	3	0.0014	360	-1	0.8413
330	2	0.0228	370	-2	0.9772
340	1	0.1587	380	-3	0.9986

将概率 p 对 x 作图（图 3-2），可得抽检特性曲线，简称 OC 曲线。

由表（3-2）可见，含量为 340g·L^{-1}的原料有 15.9%的概率被接受，如果希望接受这样的原料的概率降低至 2%，可采用两个方法。第一个办法是提高接受标准，从标准正态分布表可知，$p=2\%$时，u 值应取 2.05，代入 u 的表达式可求出 x_0（使对应于 $p=2\%$的 x 为 340），x_0 应为 360g·L^{-1}，即将接受标准改为 360g·L^{-1}，在原有分析方法的误差不变的条件下，实际含量为 340g·L^{-1}的原料被接受的概率将降至 2%。另一个办法是改善标准差 σ，因 σ 是采样误差和分析误差之和，与采样及测定次数有关，即 $\sigma=\sigma'/\sqrt{n}$，分析方法一次测定标准差为 σ，本例为10g·L^{-1}，将 σ 代入式（3-31）中，可求得 n：

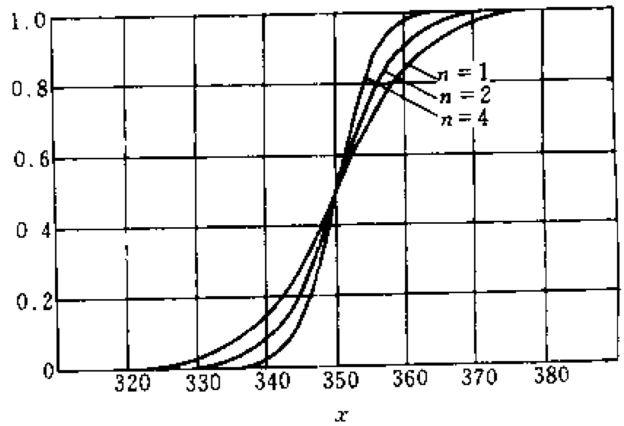


图 3-2　计量抽样检验的 OC 曲线

$$n=(2.05^2\times10^2)/(350-340)^2\approx4$$

即进行 4 次采样（分析），可在原定接受标准下，将接受实际含量为 340g·L^{-1}的原料的概率下降至 2%。

采用序贯采样法可减少检验工作量。先取少量试样作定量分析检验。如结果显示其定量指标显然在要求范围之内或之外，即可作出合格或不合格的决策；如分析结果靠近指标的边

缘，则作进一步采样检验，直至有把握作出决策。定量分析结果提供的数值信息，此时可加以充分利用，从而减少进一步的工作量（试与下面讨论的计数抽样检验比较）。

序贯检验法按如下步骤进行。取一组共 n 个试样，分析得出均值 x，将 x 值与接受/拒绝标准比较，令 x_0 为给定的生产者风险 α 值（第一类错误，以真当假，即生产者风险）对应的 x 值，x_r 为给定的消费者风险 β 值（第二类错误，以假当真，即消费者风险）对应的 x 值与 x_1 和 x_2 的关系如下：

$$x_1=n(x_0+x_r)/2+h_1 \tag{3-32a}$$

$$x_2=n(x_0+x_r)/2-h_2 \tag{3-32b}$$

式(3-32a)和式(3-32b)中

$$h_1=\sigma_s^2\ln[(1-\alpha)/\beta]/(x_0-x_r) \tag{3-33a}$$

$$h_2=\sigma_s^2\ln[(1-\beta)/\alpha]/(x_0-x_r) \tag{3-33b}$$

式中，σ_s 为表征整批试样成分变化情况的标准差，可根据原有检验记录资料计算求得。如 x 值大于 x_1 或小于 x_2，可作出判决，拒收或接受该批产品；如 x 值在 x_1 与 x_2 之间，继续采样分析，再计算累计的 x 值。用序贯抽样法可采集较少试样数（n）进行质量判决，如产品批的真值恰为 x_0，此时需采集的试样数为

$$n_{x=x}=2\sigma^2\{(1-\alpha)\ln[(1-\alpha)/\beta]-\alpha\ln[(1-\beta)/\alpha]\}/(x_0-x_r)^2$$

而 x 值为 x_r 或居于二者之间时的 n 值为

$$n_{r=x}=2\sigma^2\{(1-\beta)\ln[(1-\beta)/\alpha]-\beta\ln[(1-\alpha)/\beta]\}/(x_0-x_r)^2$$

$$n_{x=x}=2\sigma^2\{\ln[(1-\beta)/\alpha]\times\ln[(1-\beta)/\alpha]\}/(x_0-x_r)^2$$

这几个 n 均小于一次抽样检验所需要的采样数。

二、计数抽样检验

这种检验方法多用于以件计数的产品检验，如采用化学分析方法，则是作定性分析提供检验对象是否合格的属性，例如用看谱镜确定某种杂质存在或不存在。由于这里只有一比特信息量（合格或不合格两种可能性），较定量分析提供的信息量低，这种情况的采样检验需用不同的统计方法处理。

对于一批共 N 个试样，如其中不合格的占 p，合格的占 q，取 n 个试样找到 c 个不合格试样的概率 P_c 可按超几何分布计算为

$$P_c=\binom{pN}{c}\binom{N-pN}{n-c}\Big/\binom{N}{n}$$

而采集 n 个试样检验，如规定其中只有 a 个不合格时可接受，则被接受的概率为

$$P_a=\sum_{c=0}^{a}P_c=\sum_{c=0}^{a}\left[\binom{pN}{c}\binom{N-pN}{n-c}\Big/\binom{N}{n}\right]$$

当 n 值很大时上式计算量大，即 $N\to\infty$时，可用二项分布：

$$\lim_{N\to\infty}P_c=\lim_{N\to\infty}\left[\binom{pN}{c}\binom{N-pN}{n-c}\Big/\binom{N}{n}\right]$$

$$=\binom{n}{c}p^c(1-p)^{n-c}$$

一般当 N 足够大(例如 $N>10n$)时，即可用上式，此时 P_a 可计算为

$$P_a = \sum_{c=0}^{a} P_c = \sum_{c=0}^{a} \binom{n}{c} p^c (1-p)^{n-c}$$

如除 N 足够大外，同时还满足 p 较小(例如≤0.1)的条件，则可用更简洁的 Poisson 分布作近似计算：

$$P_a = \sum_{c=0}^{a} [(np)^c / c!] e^{-np} = 1 - \sum_{c=0}^{a} [(np)^c / c!] e^{-np}$$

以 P_a 对批中不合格的百分数作图，可得 OC 曲线，这种计算结果示于表 3-3。

表 3-3　产品不合格率与接受概率的关系

批不合格率 p	采集样本 n 个中所含不合格品数			接受概率 P_a		
	$n=50$	$n=100$	$n=500$	$n=50$ $a=1$	$n=100$ $a=2$	$n=500$ $a=10$
0.5	0.25	0.5	2.5	0.974	0.986	1.000
1.0	0.5	1.0	5.0	0.910	0.920	0.986
1.5	0.75	1.5	7.5	0.827	0.809	0.863
2.0	1.0	2.0	10.0	0.736	0.677	0.583
2.5	1.25	2.5	12.5	0.645	0.597	0.297
3.0	1.5	3.0	15.0	0.558	0.423	0.118
4.0	2.0	4.0	20.0	0.406	0.238	0.011
5.0	2.5	5.0	25.0	0.288	0.125	0.001
10.0	5.0	10.0	50.0	0.040	0.003	0.000

由表 3-3 可以看出，如制定的抽样检验方案是取 50 个样检验，出现 1 个或无不合格品时接受该批试样，这并不意味着产品的不合格率一定是小于 2%，不合格率为 4%的产品批，仍有 40%的概率被接受，这就是消费者的风险。反之，1%不合格率的产品，有（1－0.910）即 9%的概率被拒收，这就是生产者的风险。要完全消除这两者风险，只有进行试样的全部数检，而这需要耗费很大的人力和物力。序贯抽样检验能减少达到相同的风险水平需要的采样个数。设 a_a 是与给定生产者风险（$1-\alpha$）对应的整批产品应予接受时的不合格产品百分数，这个可接受质量水平称 AQL（Acceptable Quality Level）。a_r 是与消费者风险 β 相对应的整批产品应予拒收时的不合格产品百分数，这个应予拒收的质量水平称批不合格品百分数的允许界限 LTPD（lot tolerance percent defective）。今取 n 个试样检验，如发现有 a 个不合格，将 a 与 a_1、a_2 比较：

$$a_1 = -h_1 + sn \tag{3-34a}$$

$$a_2 = h_2 + sn \tag{3-34b}$$

式（3-34）中

$$s = (a_r - a_a) / \ln(a_r / a_a)$$

$$h_1 = \ln[(1-\alpha)/\beta] / \ln(a_r / a_a)$$

$$h_2 = \ln[(1-\beta)/\alpha] / \ln(a_r / a_a)$$

如 $a < a_1$，接受该批产品；如 $a > a_2$，拒收该批产品；如 a 在 a_1 与 a_2 之间，继续采样并重新计算 a_1、a_2 进行判决。

对质量恰符合 AQL 的一批产品，采样数的期望值为

$$n(a_a) = [(1-\alpha)h_1 - \alpha h_2] / (s - a_a) \tag{3-35}$$

对质量恰为 LTPD 的一批产品，则有

$$n(a_r)=[(1-\beta)h_2-\beta h_1]/(a_r-s) \tag{3-36}$$

采样数的期望值最大为(h_2h_1/s)，此时质量水平趋近于s。

表3-3的数据表明，如果不用序贯采样法，取500个样检验，当不合格品大于10时拒收，则拒收含1.5%(a_a=0.015)不合格品的生产者风险α为(1－0.86)＝0.14，接受含3.0%(a_a=0.03)不合格品的消费者风险β为0.12，此时可算得h_1=2.8413，h_2=2.6531，s=0.0216，即

$$a_1=-2.8413+0.0216n$$

$$a_2=2.6531+0.0216n$$

这时采样数的最大期望值为349，即维持同样水平的生产者风险与消费者风险，在最不利的情况下亦勿需采集500个试样进行检验。按此法检验，可先采150个样检验，此时a_1=0（如无不合格品，则接受），a_2=6（如大于或等于6个不合格品，则拒收），如不合格品在a_1与a_2之间，再采集150个样，此时a_1为4，a_2为9，余类推。对质量恰符合AQL（1.5%不合格率）的产品，采样数的期望值为314，对恰符合LTPD的产品，采样数的期望值为237。在确保所需的产品质量标准下，此法显然能减少检验工作量，降低生产成本。

采用科学的检验方法，是发达国家如日本确保其产品质优（退货率低，即β低），且成本合理（尽可能低的α）的诀窍。而有的生产部门由于分析检验人员缺乏统计采样知识，不能理解或拒不使用本章论及的采样方法，沿用一些缺乏科学根据的采样方法，例如“固定比例抽检法”，导致产品质量长期不能改善（β高）或生产成本高（α高）。显然，普及有关采样理论知识，具有重要的实际意义。

参 考 文 献

1 Yu R Q. Encyclopedia of Analytical Science, 1994;4518～4525

2 Youden W J. J Assoc of Anal Chem, 1967; 50: 1007

3 Ingamell C O. Tananta, 1974; 21:141～155

4 Kratochvil B, Taylor J K. Anal Chem, 1981; 53: 954A

5 Laitinen H A, Harris W E. Sampling. Chemical Analysis, 2nd edn, New York: McGraw-Hill, chap. 27, 1975, p565～582

6 Smith R, James G V. The Sampling of Bulk Materials, Royal Society of Chemistry, London: Burington House, 1981

7 Taylor J K. Quality Assurance of Chemical Measurements, Lewis:Michigan, 1987; p. 55～74

8 Kateman G, Pijpers F W. Quality Control in Analytical Chemistry, New York:John Wiley & Sons, 1981

9 Kowalski B R, Reidel D. Chemomtrics. Mathematics and Statistics in Chemistry. Dordrecht:pub. Co., 1983. p. 177

10 Muskens P J W M, Kateman G. Anal Chim Acta, 1978; 103: 1

11 Kateman G, Muskens P J W M. Anal Chim Acta, 1978; 103: 11

12 Muskens P J W M. Anal Chim Acta, 1978; 103: 445

第四章　化学试验设计与优化方法

第一节　概　　述

本章介绍分析化学中试验设计与优化方法，第一部分以目前统计学中最重要的实验设计体系，即因子设计及部分因子设计、正交试验设计、均匀试验设计、最优设计为主要对象来介绍该研究领域的主要成果。混料试验设计为上述试验设计的一种特殊情况，具有其特殊性，故也分出一节来专门介绍。对于已为广大化学计量学家熟悉的单纯形方法亦将给出必要的介绍。对于部分因子设计、正交试验设计和均匀试验设计中的一些必要的表格也将给出，以备读者使用。第二部分以目前应用数学中发展的优化算法为主线，特别对化学计量学中常用优化方法给出较详细的介绍。介绍将分局部优化算法和全局优化算法两部分来进行阐述。在介绍经典的局部最优方法方面，以在化学中得到广泛运用的方法为主；在介绍全局最优方法方面，首先将目前在化学计量学研究中备受关注的随机寻优法，即模拟退火算法和遗传算法给出较详细的介绍，然后对伪蒙特卡罗法，即基于数论方法的序贯优化法也作出介绍。对这三种方法都将给出其计算机程序，供读者查考。

从目前的情况看，整个化学学科实际上还是一门以实验研究为主的自然科学的分支学科，所以有关如何通过试验设计的方法来最有效地获取相应的化学信息，以达到最优进行化学反应的条件应是化学乃至化工研究中的一个十分重要的问题。就分析化学而言，通过试验设计以找到最优的量测实验条件一直是化学计量学研究的一个重要内容。如色谱分析中的最优分离条件的选择、各种仪器分析方法中的诸多参量的选择等。可以这么说，对于化学量测实验，试验设计的成败，关系到实验能否得到包含所需信息的化学量测数据，一个失败的试验设计将导致所得到的数据中包含的信息量极低，任何卓有成效的处理数据的化学计量学方法也无法从中提取有用的信息。在系统讨论试验设计方法之前，在此将这一领域的基本概念及常用术语给出必要的介绍。

在试验设计中，衡量试验效果的量称为试验指标（简称指标）或响应值，能用数值表述的指标称定量指标，如化学反应的产率、分析试验的检测限或其他品质因数等。亦有不是用量表示的指标，如化学产品的色度等，这种指标称为定性指标。定性指标常可转化为定量指标，如用 5 级计分进行评分等。当试验设计的指标要用一组数表示时，如分析方法的优化需考虑灵敏度、准确度、选择性等，称多指标试验设计的问题。

完成一项试验，得到指标的试验观测值，以 y 表示。当改变试验条件时，指标 y 将发生变化，影响 y 取值的量称为因素（亦称因子），因素也可以是定量因素或定性因素，和指标一样，定性因素总是可转化为定量因素。因素在试验中的状态，例如定量因素的取值，称为因素（或因子）的水平。一般试验方案是由若干个试验组成，因素在这些试验中变化了几种状态就称为几种水平。在试验中，有的因素所处的状态是不可控制的，例如在自然条件下进行的某些试验，但一般化学实验的条件多是可以控制的，本章将主要研究可控因素。在试验设计中，只对可控因素在试验前作出设计，而对不可控因素，则在试验过程中记录其水平，在数据分析中加以处理。

根据影响指标的因素的多少，试验可分为单因素试验和多因素试验，化学试验包括分析化学量测试验一般均是多因素试验，用 x_1、x_2，…，x_n，表示试验中的 n 个因素，这 n 个因素的可能取值的区域称为试验域，记为 $\Re$，y 可看成定义在 $\Re$ 上的一个函数

$$y=f(x)$$

在此，$x=[x_1,x_2,\cdots,x_n]^t$。当函数 $f(.)$已知时，用最优试验设计方法，如 D 最优设计等当是最好的选择，但多数化学试验中，函数 $f(.)$不能完全明了，这时无法直接写出 $f(.)$的数学表达式。一种方法是通过大量试验构造一类函数来逼近这些试验数据，然后再从函数求最优解，并通过试验验证。另一种方法是不研究试验指标与因子之间的函数关系，而只寻求试验指标最优的诸因素取值，这种方法称为“黑箱”式方法。

所谓试验设计，归根结底是如何在试验域上最有效地选择试验点，通过试验得到指标(或响应)的观测值，然后进行数据分析求得指标(或响应)取最优值的条件。因此，试验设计的目标，就是要用较少的试验取得关于 $f(x)$尽可能充分的信息。

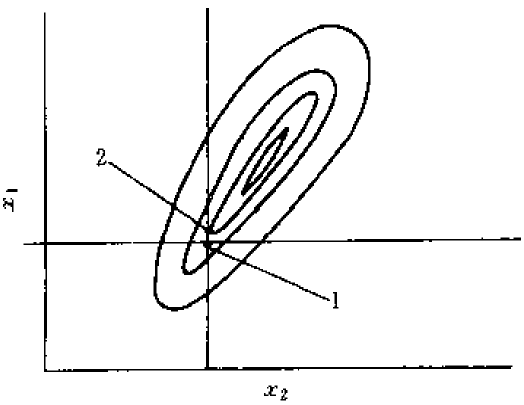


图 4-1　固定其他变量变动一个变量的寻求最优实验条件所得结果示意图
1　固定 x_1，变动 x_2 找到的第一个最优点；
2—固定 x_2，变动 x_1 找到的第二个最优点

虽然在化学实验研究中大多是多变量体系，可是、化学文献特别是分析化学文献中许多有关试验的报道本质上是按单因素试验设计的，即通常在化学试验中采用的每次只变动一个因素而固定其他因素的设计方法，这种方法当然是最简单和最直观的，然而，一般说来，对于绝大多数多因素化学体系，用这种方法来设计试验，是很难找到真正的最优实验条件的。在许多情况下，得到的结果并不是最优的因子水平(参见图4-1)。

在试验设计和优化中，还可区分同时试验与序贯试验两种工作方法。所谓同时试验，就是通过试验设计对有关因素的水平进行规划后，同时进行诸因子各水平的试验，然后综合分析得到的试验结果，求出最优条件。另一种方法是每进行一次或少数几次试验后，先分析已取得的试验结果，再根据这些结果规划下一步的试验，这样序贯试验向最优条件逼近。目前应用广泛的正交试验或称为部分因子试验、均匀试验设计及最优试验设计基本上属于同时试验法，而序贯试验法的典型代表是单纯形优化法。同时试验与序贯试验可在优化试验中综合使用。

第二节　因子设计及析因分析方法

所谓因子设计实际上是一种多因素试验设计方法[1~4]，它通常可由多种因素的水平数的一个乘积表来表示。如一个三因子二水平的因子设计可由 2^3 来表示，亦即 $2\times2\times2$，需进行 8 次试验；而一个四因子三水平的因子设计则可由 3^4 来表示，亦即 $3\times3\times3\times3$，需共进行81 次试验。因子设计的任务就是要通过这样的试验安排来了解各个因素及各因素水平之间的搭配对响应值或指标的影响，即析因问题。故析因问题是因子设计的主要研究的问题之一。下面我们将通过一个具体的二水平三因子试验设计的例子来具体说明因子设计中析因分析的主要思路和方法。

以用反向高效液相色谱分离酚类为例。在此，主要考察三个因素：① 流动相中甲醇的比例(流动相：水和甲醇)为 70%与 80%；② 乙酸的浓度($mol\cdot L^{-1}$)为 0.004 和 0.01；③ 柠檬酸浓

度(g·L⁻¹)为 2 和 6。由此可以得到由表 4-1 给出的一个试验设计表，在此，响应值色谱分离度是一个判别整个分析过程酚类物质分离程度的一个度量。其值越大说明分离效果越好。

表 4-1　反向高效液相色谱分离酚类的因子设计表(2^3)

试验编号	乙酸浓度① (A)	甲醇比例② (M)	柠檬酸浓度③ (C)	色谱分离度响应值
1	−	−		10.0
2	+	−		9.5
3		+	−	11.0
4	+	+	−	10.7
5	−	−	+	9.3
6	+	−	+	8.8
7	−	+	+	11.9
8	+	+	+	11.7

① 乙酸浓度(Acetic acid,A),(单位为 mol·L⁻¹):0.004(−)和 0.01(+)。

② 甲醇比例(Methol,M)(流动相:水和甲醇):70%(−)与 80%(+)。

③ 柠檬酸浓度(Citric acid,C),(单位为 g·L⁻¹):2(—)和 6(+)。

一、主效应的估价

在上述试验中，我们最先想知道的当然是每一种因素对分离的贡献，也就是说每种因子对于试验效果的主效应。从表中我们可以看到，对于乙酸而言，试验序号 1 和 2 的差别就在于乙酸，别的条件都一样。同理，试验序号 3 和 4、5 和 6、7 和 8 对于乙酸而言也都是一样，所以，要求乙酸对分离效果的主效应可通过下式求得

$$\begin{aligned}\text{主效应}_{(\text{乙酸})} &= [(y_2-y_1)+(y_4-y_3)+(y_6-y_5)+(y_8-y_7)]/4 \\ &= [(9.5-10.0)-(10.7-11.0)+(8.8-9.3)+(11.7-11.9)]/4 \\ &= -0.375\end{aligned}$$

因此，将乙酸浓度从 0.004 增加到 0.01(mol·L⁻¹)将对分离效果产生负作用。同理，可以通过表 4-1 分别算出甲醇和柠檬酸的主效应

$$\begin{aligned}\text{主效应}_{(\text{甲醇})} &= [(y_3+y_4+y_7+y_8)-(y_1+y_2+y_5+y_6)]/4 \\ &= [(11.0+10.7+11.9+11.7)-(10.0+9.5+9.3+8.8)]/4 \\ &= 1.925\end{aligned}$$

$$\begin{aligned}\text{主效应}_{(\text{柠檬酸})} &= [(y_5+y_6+y_7+y_8)-(y_1+y_2+y_3+y_4)]/4 \\ &= [(9.3+8.8+11.9+11.7)-(10.0+9.5+11.0+10.7)]/4 \\ &= 0.125\end{aligned}$$

从上述计算可以看出，根据表 4-1 的数据，对于每一个因素，只要将正号的响应值加合起来减去带负号的响应值之和，就得到了每个因素的主效应。值得指出的是，对于这样的 2^3 因子设计，如图 4-2 示出的立体直观图是十分有用的，从图我们就可直观地看出各因素的主体效应。

二、交叉效应的估价

如果所研究的对象为一线性模型，则主效应的计算就说明了它们本身对于模型影响的大小，可是，在绝大多数的实际问题中，都存在着所谓的交叉效应，所以，在估价了它们自身的主效应之后，一般都需要继续估价交叉效应。

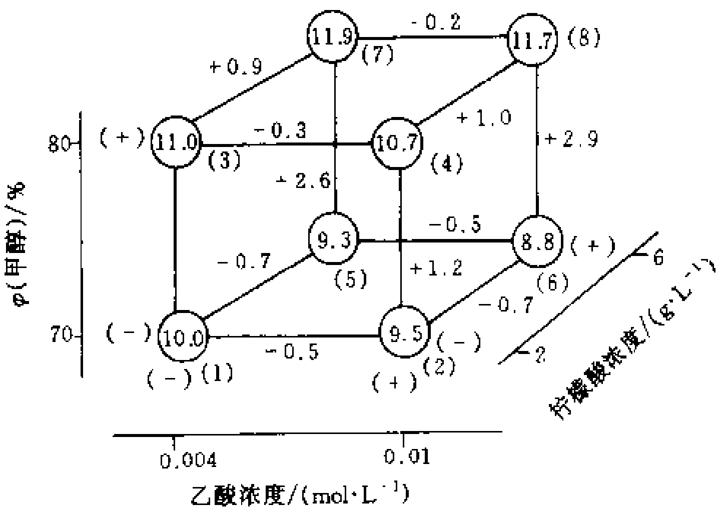


图 4-2　2^3 因子设计的立体直观效应图

首先，是定性识别是否存在交叉效应。我们只需将两个不同因素对响应值作图，就很容易看出该两因素是否存在着交叉效应。图 4-3 就给出了这样一个采用作图方法来鉴别是否存在交叉效应的例子。从图 4-3 所示的情况不难理解，如果图中的两条直线平行，则说明第二个因素变化对第一个因素的主效应没有影响，也就是说，第二个因素的变化引起的响应值的变化与第一个因素变化时对响应值的影响是可以线性叠加的；反之，如果图中的两条直线不平行，则说明在第二个因素发生变化时，第一个因素的变化将或是加强或是减弱第二因素对响应值的效应，这正说明了该两因素存在交叉效应。这样的交叉效应在化学化工的研究实践中是很常见的。

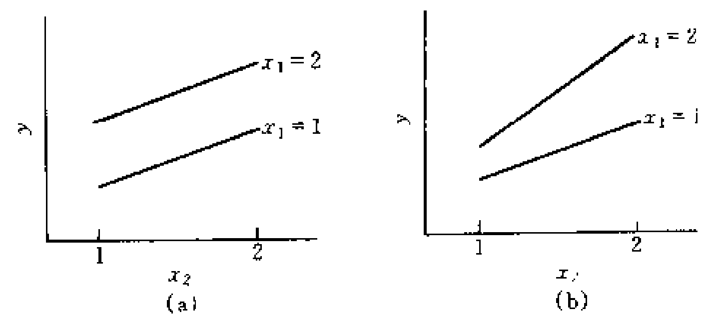


图 4-3　是否存在交叉效应的示意图

(a) 不同 x_1 两线平行，说明不存在交叉效应；(b) 不同 x_1 两线不平行，说明存在交叉效应

从图 4-2 还可以很清楚地看出不同水平的响应值的差别。对于柠檬酸和甲醇的交叉效应，也可以从图中看出，这是因为对于甲醇效应在柠檬酸浓度为 6g・L^{-1}时要明显大于在柠檬酸浓度为 2g・L^{-1}时效应，这就说明甲醇效应和柠檬酸的效应不具有线性加合性，它们之间存在着交叉效应。此交叉效应可通过以下步骤来进行具体计算。先估价柠檬酸不同水平的甲醇平均效应，即当柠檬酸为 6g・L^{-1}时，甲醇的平均效应为$[(y_8-y_6)+(y_7-y_5)]/2=(2.9+2.6)/2=2.75$；而当柠檬酸为 2g・$L^{-1}$时，甲醇的平均效应为$[(y_4-y_2)+(y_3-y_1)]/2=(1.2+1.0)/2=1.1$；然后，将所得两数相减，继续除以 2，即得到了甲醇对柠檬酸的交叉效应值，即

$$
\begin{aligned}
\text{交叉效应}_{(\text{甲醇对柠檬酸})} &= \{[(y_8-y_6)+(y_7-y_5)]/2-[(y_4-y_2)+(y_3-y_1)]/2\}/2 \\
&= \{(2.9+2.6)/2-(1.2+1.0)/2\}/2 \\
&= (2.75-1.1)/2=0.825
\end{aligned}
$$

甲醇与柠檬酸的交叉效应应与柠檬酸与甲醇的交叉效应相同，所以，对于它们的交叉效应还可以从先估价甲醇不同水平的柠檬酸平均效应，再按上述过程进行计算，即

$$\begin{aligned}交叉效应_{(柠檬酸对甲醇)} &= \{[(y_8-y_4)+(y_7-y_3)]/2-[(y_5-y_2)+(y_5-y_1)]/2\}/2\\ &=\{(1.0+0.9)/2-(-0.7\quad 0.7)/2\}/2\\ &=(0.95+0.7)/2=0.825\end{aligned}$$

如果将以上计算式修改一下，可以得到

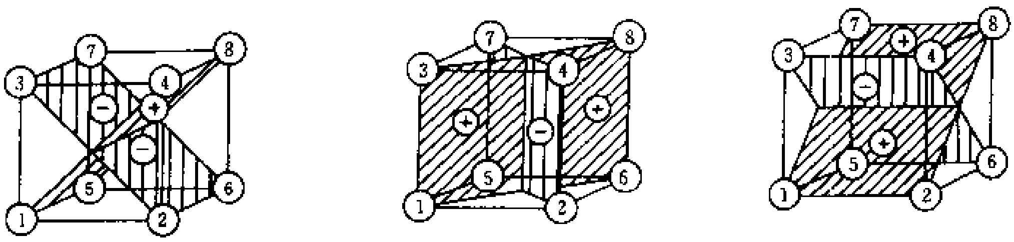
$$\begin{aligned}交叉效应_{(柠檬酸对甲醇)} &= \frac{y_8+y_7+y_2+y_1}{4}-\frac{y_6+y_5+y_4+y_3}{4}\\ &=0.825\end{aligned}$$

这样，对于交叉效应同样可以通过如图 4-2 的立体直观图来表示，对于交叉效应的立体直观图示于图 4-4。图 4-4 由三个子图给出，最上面的图，即图 4-4(a)，则表示了乙酸和甲醇的交叉效应，其中由带(+)号的平面所连的 4 个顶点，表示在计算中它们的符号都取正值，而由带(—)号的平面所连的 4 个顶点则取负值，所以，如果要计算乙酸和甲醇的交叉效应，就可直接用下式进行计算：

$$\begin{aligned}交叉效应_{(乙酸对甲醇)} &= \frac{y_8+y_5+y_4+y_1}{4}-\frac{y_7+y_6+y_3+y_2}{4}\\ &=\frac{11.7+9.3+10.7+10.0}{4}-\frac{11.9+8.8+11.0+9.5}{4}\\ &=0.125\end{aligned}$$

同理，也可以参照图 4-4(b)来计算乙酸和柠檬酸的交叉效应，即

$$\begin{aligned}交叉效应_{(乙酸对柠檬酸)} &= \frac{y_8+y_6+y_3+y_1}{4}-\frac{y_7+y_5+y_4+y_2}{4}\\ &=\frac{11.7+8.8+11.0+10.0}{4}-\frac{11.9+9.3+10.7+9.5}{4}\\ &=0.025\end{aligned}$$



(a) (b) (c)

图 4-4 交叉效应立体直观图

图中顶点编号同图 4-2

(a) 乙酸(A)和甲醇(M)的交叉效应；(b) 乙酸(A)和柠檬酸(C)的交叉效应；(c) 甲醇(M)和柠檬酸(C)的交叉效应

同时，还可以用交叉效应立体直观图来计算乙酸、甲醇和柠檬酸的三项交叉效应。在图 4-4 (a) 中，当柠檬酸的水平为 6g·L^{-1}时，甲醇和乙酸交叉效应的影响可由下式给出

$$\frac{(y_8-y_7)-(y_6-y_5)}{2}=\frac{(11.7-11.9)-(8.8-9.3)}{2}=0.15$$

而当柠檬酸的水平为 2g·cm^{-3}时，甲醇和乙酸交叉效应的影响则由下式给出

$$\frac{(y_4-y_3)-(y_2-y_1)}{2}=\frac{(10.7-11.0)-(9.5-10.0)}{2}=0.1$$

所以，乙酸、甲醇和柠檬酸的三项交叉效应为

$$\frac{0.15-0.1}{2}=0.025$$

同样，也可以由图 4-4（b）或图 4-4（c）来计算三项交叉效应，所得结果一样。

另外，还可直接用表 4-1 给出的因子设计表（2^3）来得到交叉项计算的符号。算法很简单，只需将要计算的交叉项在表中的符号相乘，根据正正得正、负负得正和正负得负的原则，即可计算出所有交叉项的符号来。表 4-2 给出了据此原则得出的 2^3 因子设计表的全部主项及交叉项计算表，只需根据此表就可容易地得到计算各种效应所需的符号来。

表 4-2　2^3 因子设计的全部主项及交叉项计算符号表[①]

编号	*A*	*M*	*C*	*AM*	*AC*	*MC*	*AMC*	色谱分离度
1	−	−	−	+	−	+		10.0
2	+	−	−	−	−	+	+	9.5
3	−	+		−	−	−	+	11.0
4	+	+	−	+	−			10.7
5	−	−	+		−	−	−	9.3
6	+	−	+	−	+	−	−	8.8
7	−	+	+			+	−	11.9
8	+	+	+	+	+	+	+	11.7

注：*A*——乙酸浓度（$mol\cdot L^{-1}$）：0.004（−）和 0.01（+）；

M——甲醇比例（流动相：水和甲醇）：70%（−）与 80%（+）；

C——柠檬酸浓度（$g\cdot L^{-1}$）：2（−）和 6（+）；

AM，*AC*，*MC*，*AMC*——两个或三个因素的交叉效应。

以上情况虽只是以 2^3 因子设计为例来进行说明，其实，其基本原理将对别的因子设计的情况都是适用的。

三、效应及残差正态图

以上讨论了因子设计的主效应及交叉效应的计算，实际上，还需对这些效应进行统计估价，即决定哪些效应将在模型建立时需要考虑，哪些则可以忽略，效应正态图即可以用来进行这样的决断；而残差正态图则可用来进行判断所得模型是否合理。

因为正态分布一般表示为由多种不定因素而产生出来的一种分布，所以，如果某些效应是服从正态分布的话就可认为它们实际对实验不产生显著影响。用以检测一系列变量是否服从正态分布的最简易方法就是采用正态分布图来对它们进行检验。正态分布图的构造可由以下步骤来完成：

① 先将要检验的一系列变量按大小进行排列，对于已得到的各种效应可得如表 4-3 所示的排列；

② 计算累积概率：对于有 T 个数据的系列，可根据以下公式来计算它们的累积概率

$$P_i=\frac{i-0.5}{T}\times 100\% \tag{4-1}$$

式中，P_i——累积概率，%。

③ 以需检验变量的标度为 x 轴，以累积概率为 y 轴作图，在图上能用一条直线描述的变量可视为是服从正态分布的变量。

表 4-3 色谱分离试验所得各种效应的顺序排列表①

效应名称	A	AC	AMC	C	AM	MC	M
效应数值	−0.375	0.025	0.025	0.125	0.125	0.825	1.925
概　率	7.14	21.43	35.71	50	64.88	78.57	92.86

① 此表所用符号同表 4-2。

由表 4-3 所示的色谱分离试验所得各种效应的正态分布图示于图 4-5，从图可以看出，效应 AC（乙酸与柠檬酸的交叉效应）、AMC（甲醇、乙酸与柠檬酸的三项交叉效应）、C（柠檬酸的主效应）以及 AM（乙酸与甲醇的交叉效应）正好落在一条直线上，说明它们对试验的影响很小，可以忽略。于是，如果需要对此色谱分离试验建立回归模型的话，就只需选择乙酸（A）、甲醇（M）和甲醇与柠檬酸的交叉效应（MC）来建立相应的模型就可以了。经对数据建模可得

$$y=10.363-(0.375/2)x_A+(1.925/2)x_M+(0.825/2)x_{MC}$$

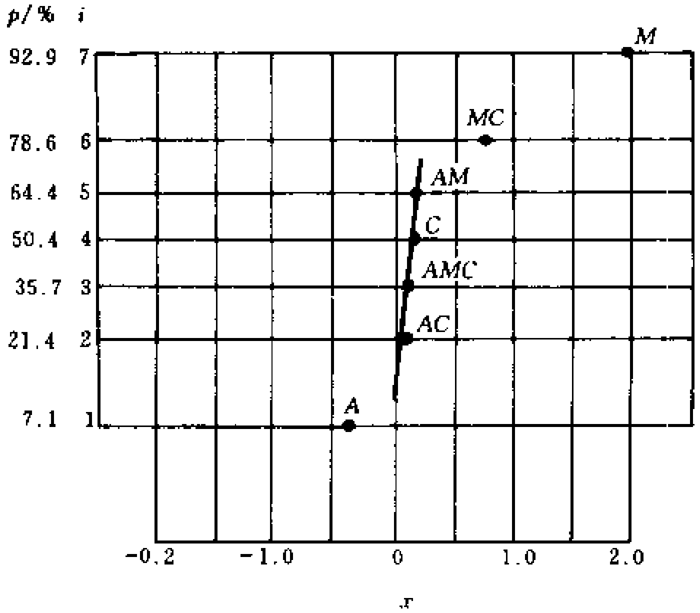


图 4-5 色谱分离试验的各种效应正态分布图

得到上述回归模型后，可以用它来计算该模型的残差，如果所得残差按上述方法所得的残差正态分布图可用一条直线表示的话，说明模型是很合理的。用此模型算出的残差列于表 4-4，它们的残差正态分布图示于图 4-6。这些残差点似可通过一条直线表示出，但吻合不是很好。

表 4-4 色谱分离试验所得残差表

编　号	1	2	3	4	5	6	7	8
实验值	10.0	9.5	11.0	10.7	9.3	8.8	11.9	11.7
计算值	10.0	8.80	11.10	11.55	10.0	8.80	11.10	11.55
残　差	−0.0	0.70	−0.10	−0.85	−0.70	−0.00	0.80	0.15

从上述讨论可知，因子设计的确具有不少优点，它不但可方便地估价出各种因素的主效应和它们之间的交叉效应，而且还可方便地建模，并对模型进行相应的估价。可是在化学化工的实际研究中，像因素只有 3 个的并不多，而且因素试验的水平也不可能只限于 2，只要因素和水平数一增加，因子设计就显出了它的不足。即使全部因素均是二水平的，当因素数为 n 时，总试验数就是 $N=2^n$。化学化工的诸多研究常涉及很多因素，例如多达 5～10 个因素时，

此时如采用因子设计，试验数就会太多，如 5 个因素的话就需要 32 次试验，如果是 10 个因素的话，试验次数可达 $N=2^{10}=1024$ 次！而且，对于这样的因子设计，其交叉效应的计算也显著增加，表 4-5 列出了当因素为 2～8 时，各主效应和交叉效应所需计算的次数，就充分说明因子设计在因素多于 5 时，使用起来就很困难了。这样，就提出了部分因子设计和正交设计的方法来解决这样的问题。另外，如果因子的水平数很高时，就用正交设计也会感到试验次数太多，于是就产生了均匀设计方法。

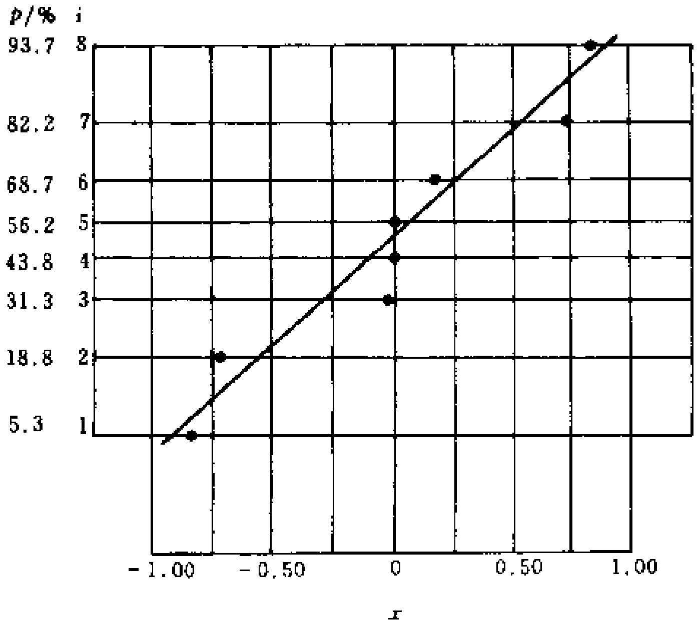


图 4-6 色谱分离试验的残差正态分布图

表 4-5 2^n 因子设计时需计算的效应数①

效　　应	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$
主效应	2	3	4	5	6	7	8
2 因素交叉	1	3	6	10	15	21	28
3 因素交叉		1	4	10	20	35	56
4 因素交叉			1	5	15	35	70
5 因素交叉				1	6	21	56
6 因素交叉					1	7	28
7 因素交叉						1	8
8 因素交叉							1

① p 次交叉效应的数目计算公式为：$m_p=C_n^p=n!/[p!(n-p)!]$。

第三节　部分因子设计

对于部分因子设计，2 水平的设计方法比较成熟，在此仅对它们如何构建及其有关的一些重要理论和概念进行必要的介绍。对于 2 水平的部分因子设计，一般都具有 2^{n-d} 的形式，其中 2 表示因子的水平数，n 为因子数（或称因素的数目），d 为可能缩小试验次数的 2 的乘方数，当 d 为 1 时，因试验次数可以缩小一半，故有半因子设计法之称；当 d 为 2 时，试验次数可以降至全因子设计所需数目的 1/4，故有四分之一因子设计法之称。

一、半因子设计法

所谓半因子设计，实际上就是 2^{n-1} 次因子设计，在此 n 为因子数，2^{n-1} 为其试验次数，因其试验次数实际上只有全因子设计的一半，故有半因子设计之称。在此我们仍以一个实际例

子来进行说明。这是一个关于磷酸酶活性的例子。

从以前的化学实验，知道磷酸酶活性与硫酸锌（简记为 Z）、硫酸镁（简记为 M）、pH 值（简记为 P）、对硝基苯基磷酸二钠（简记为 D）及 2-氨基-2-甲基-1-丙醇（简记为 A）这五个因素有关，所以，可以构造一个五因素两水平的全因子设计表来进行实验，以得到有关磷酸酶活性的主要影响因素等信息。有关这五个因子的两个水平示于表 4-6。

表 4-6　影响磷酸酶活性的 5 个因子的两水平说明表

影响因素	单　位	水　平	
		−	+
Z	$\mu mol \cdot dm^{-3}$	40	80
M	$\mu mol \cdot dm^{-3}$	1.50	2.50
P	1	10.00	10.70
D	$mmol \cdot dm^{-3}$	10	20
A	$mol \cdot dm^{-3}$	0.20	0.60

由 2^5 次全因子设计所得结果示于表 4-7。

表 4-7　磷酸酶活性 2^5 次全因子设计结果

实验号	因　子					酶活性/u①	实验号	因　子					酶活性/u①
	Z	M	P	D	A			Z	M	P	D	A	
1	−	−	−	−	−	109	17*	−	−	−	−	+	106
2*	−	−	−	−	−	113	18	+	−	−	−	+	120
3*	−	+	−	−	−	103	19	−	+	−	−	+	113
4	+	+	−	−	−	113	20*	+	+	−	−	+	115
5*	−	−	+	−	−	103	21	−		+	−	+	109
6	+	−	+	−	−	104	22*	+	−	+	−	+	117
7	−	+	+	−	−	106	23*	−	+	+	−	+	105
8*	+	+	+	−	−	123	24	+	+	+	−	+	115
9*	−	−	−	+	−	119	25		−	−	+	+	96
10	+	−	−	+	−	146	26*	+	−	−	+	+	128
11	−	+	−	+	−	111	27*	−	+	−	+	+	95
12*	+	+	−	+	−	143	28	+	+	−	+	+	127
13	−	−	+	+	−	116	29*	−	−	+	+	+	99
14*	+	−	+	+	−	145	30	−	−	+	+	+	131
15*	−	+	+	+	−	110	31	−	+	+	+	+	92
16	+	+	+	+	−	148	32*	−	+	+	+	+	132

① $1u = 1\mu mol \cdot min^{-1} = 16.67 nmol \cdot s^{-1}$。

按上节所讨论的方法，可以很容易地算出所有的主效应和它们的交叉效应，这些主效应和交叉效应的具体数值列于表 4-8。

表 4-8　磷酸酶活性 2^5 次全因子主效应和交叉效应结果

因　子	估计效应	因　子	估计效应	因　子	估计效应	因　子	估计效应
	116.00	ZP	1.38	ZMP	2.25	PDA	0.88
		ZD	12.25	ZMD	0.63		
Z	20.50	ZA	0.75	ZMA	−2.38	$ZMPD$	−0.75
M	−0.63	MP	1.50	ZPD	0.63	$ZMPA$	0.50
P	−0.13	MD	−2.13	ZPA	−0.13	$ZMDA$	1.63
D	10.25	MA	−0.88	ZDA	0.50	$ZPDA$	0.13
A	−7.00	PD	1.13	MPD	−1.00	$MPDA$	1.50
		PA	0.13	MPA	−3.00		
ZM	2.13	DA	−10.25	MDA	1.63	$ZMPDA$	0.00

采用这些计算所得的主效应和交叉效应的数值，可得如图 4-7 所示的效应正态分布图。

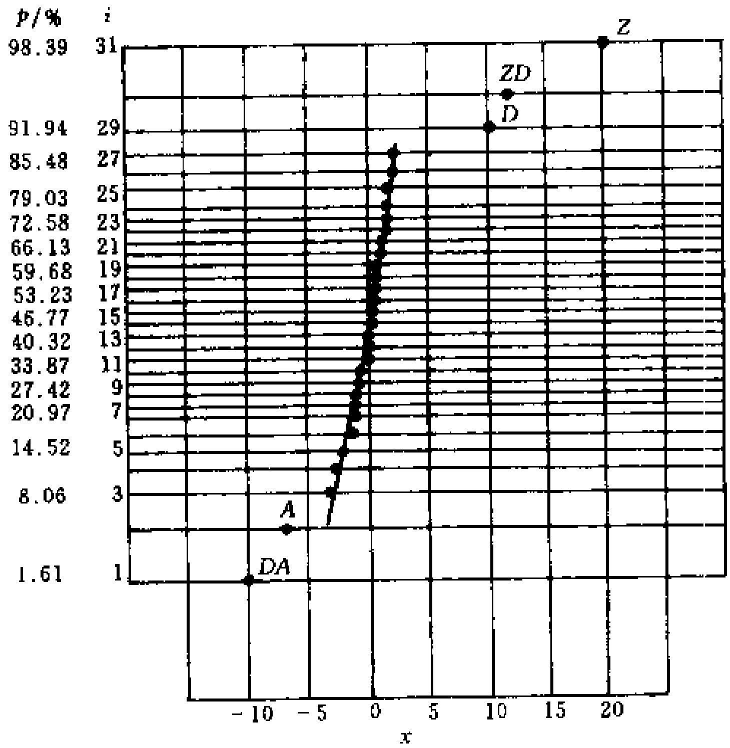


图 4-7 磷酸酶活性 2^5 次全因子效应正态分布图

从图 4-7 可以很清楚地看出，只有 *A*（2-氨基-2-甲基-1-丙醇）、*D*（对硝基苯基磷酸二钠）和 *Z*（硫酸锌）这三个主因素及 *ZD*（对硝基苯基磷酸二钠和硫酸锌的交叉效应）、*DA*（2-氨基-2-甲基-1-丙醇与对硝基苯基磷酸二钠的交叉效应）才对磷酸酶活性有显著影响，其他别的主效应和交叉效应都可以忽略。这个结果似乎说明对于磷酸酶活性的试验原本就可只用四因子设计表来完成，因为pH 效应本身就不显著。问题是在实验前我们没有这样的把握。但是，这也从另一面说明 2^5 次全因子可能本身就存在信息盈余，我们完全可以想办法来减少试验的次数。这就是部分因子设计的基本道理。

表 4-7 中某些试验标记了星花（*）、现在将它们抽出以作为因子设计表，并只用它们来进行主效应和交叉效应的计算，这样可以得到如表 4-9 所示的结果，将得到的效应正态分布图示于图 4-8，从图可以看出，所得结果完全与 2^5 次全因子效应正态分布图的结果一致，说明用部分因子设计来代替全因子设计是很有道理的。

表 4-9 由表 4-7 中标记（*）的试验结果算得各种效应值

因 子	估计效应	因 子	估计效应	因 子	估计效应	因 子	估计效应
总均值	116.00	*D*	10.75	*ZD*	9.25	*MA*	−0.25
Z	22.00	*A*	−7.75	*ZA*	−0.25	*PD*	−1.25
M	−0.50	*ZM*	3.00	*MP*	2.00	*PA*	0.75
P	1.50	*ZP*	3.00	*MD*	−2.25	*DA*	−8.00

将仔细讨论一下 2^{5-1}次半因子设计的基本原理及在部分因子设计中的十分重要的概念。用表 4-7 中标记了星花（*）的试验来组成一个 2^{5-1}次的半因子设计表，并用它们来计算需要计算的交叉项，按前节所讲的办法，将它们在表中的符号相乘，根据正正得正、负负得正和正负得负的原则，计算出所有交叉项的符号来。表 4-10 中给出了这样计算所得的部分结果。

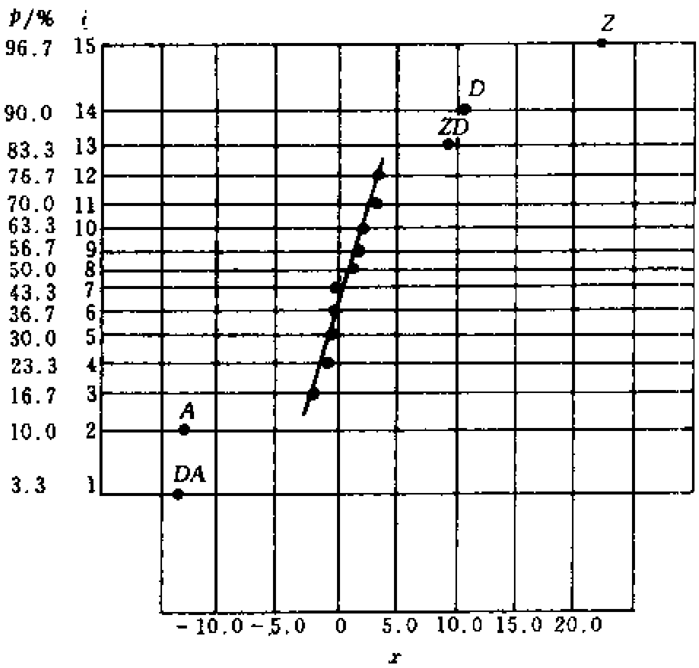


图 4-8　2^{5-1}次半因子设计的效应正态分布图

表 4-10　由 2^{5-1}次半因子设计表算得部分主效应和交叉效应的结果

Z	*MPDA*	*A*	*ZMPD*	*ZM*	*PDA*	*ZA*	*MPD*
−	−	+	+	+	+	−	−
+	+	−	−		−	−	−
−	−	−	−	−	−	+	+
+	+	+	+	+	+	+	+
−	−	−	−	+	+	+	−
+	+	+	+	−	−	+	+
−	−	+	+	−	−	−	−
+	+	−	−	+	+	−	−
−	−	−	−	+	+	+	+
+	+	+	+	−	−	+	+
−	−	+	+	−	−	−	−
+	+	−	−	+	+	−	−
−	−	+	+	+	+	−	−
+	+	−	−	−	−	−	−
−	−	−	−	−	−	+	+
+	+	+	+	+	+	+	+

从表 4-10 可以看出，主效应 Z（硫酸锌）与 4 项交叉效应 $MPDA$ 具有完全一致的符号；主效应 A（2-氨基-2-甲基-1-丙醇）与四项交叉效应 $ZMPD$ 也具有完全一致的符号；两项交叉效应 ZM 与三项交叉效应 PDA 也具有完全相同的符号等等。这种情况说明，对于 2^{5-1}次半因子设计的效应将存在重复的机会，这两种效应具有同一符号的情况，在部分因子设计中，统计学家将其称为共现现象，这些同符号的效应将称为同名或别名效应。对于用表 4-7 中标记了星花（*）的试验来组成的 2^{5-1}次的半因子设计表，算得的所有别名效应及其效应值都列于表 4-11。从表中还可以看出算得的别名效应一般都发生在主效应与四项交叉效应及二项交叉效应与三项交叉效应之间。主效应与二项交叉效应和三项交叉效应之间不存在共现现象。这样当然是比较理想的，因为一般说来，主效应和二项交叉效应都比较重要，在它们之间发生共现现象就不太好。

表 4-11 别名效应及其所算得的效应值

因 子	估计效应	因 子	估计效应	因 子	估计效应	因 子	估计效应
均值+ZMPDA	116.00	D+ZMPA	10.75	ZD+MPA	9.25	MA+ZPD	−0.25
Z+MPDA	22.00	A+ZMPD	7.75	ZA+MPD	−0.25	PD+ZMA	−1.25
M+ZMDA	−0.50	ZM+PDA	3.00	MP+ZDA	2.00	PA+ZMD	0.75
P+ZMDA	1.50	ZP+MDA	3.00	MD+ZPA	−2.25	DA+ZMP	−8.00

从以上分析可知，对于五因素的全因子设计，用半因子设计来替代，可得到与全因子设计几乎相同的结果，至此，很自然地也就产生了这样的问题：这样的半因子设计怎样构成？是不是存在多种构成半因子设计的可能？如果存在多种可能，怎样的半因子设计才是最好的？在以下的讨论中将逐一来回答上述问题。

二、半因子设计的产生方法及四分之一因子设计法

一个构成半因子设计的简便方法就是先设定一个同名和别名效应，然后就用这个别名效应来产生别的各个别名效应，以构成一个半因子设计，这样设定的别名效应就可称之为产生子。现仍以上述 2^{5-1}次的半因子设计来进行说明。

如令效应 A（2-氨基-2-甲基-1-丙醇）与四项交叉效应 $ZMPD$ 同名，则有

$$A=ZMPD \tag{4-2}$$

这样，可得到一个定义关系或定义相反

$$I=ZMPD\times A=ZMPDA \tag{4-3}$$

就利用式（4-3）这个定义关系，可产生所有其他别的各个别名效应来。如得到 Z 主效应的别名效应，则可将这个定义关系与 Z 效应相乘而得，即

$$Z\times I=Z\times ZMPDA=Z^2MPDA=MPDA \tag{4-4}$$

同理可求得各主效应的别名效应来，它们分别为 $M=ZPDA$；$P=ZMDA$；$D=ZMPA$。对于二项交叉效应也可以用上述乘法原则计算，例如

$$ZM=ZM\times ZMPDA=Z^2M^2PDA=PDA$$

别的二项交叉效应亦可仿此产生，这样，就得到了与表 4-11 完全相同的同名或别名表。

此外，如果定义 Z 与 $MPDA$ 同名，则有 $Z=MPDA$。对于 $Z=MPDA$ 这样定义的产生子，虽表面上看来与 $A=ZMPD$ 不同，但实际上它们是一样的，这是因为它们定义了同一个定义关系 I，即

$$I=Z\times MPDA=A\times ZMPD \tag{4-5}$$

所以，由这两个产生子产生出来的半因子设计实际上是完全一样的。当然，如果定义 $A=ZM$ 或 $A=ZMP$，这样就可产生不同的新半因子设计，但这就造成了主因素与二项交叉效应或三项交叉效应的同名，从而使得该部分因子设计的分辨率不高，这在下一节将给出较详细讨论。在此，如果令

$$A=-ZMPD$$

此时，$I=-AZMPD$，它确实可以产生一个与 $I=AZMPD$ 不同的半因子设计，而且，将这两个设计表合在一起，就可成为一个完整的 2^5 全因子设计表。注意考察表 4-7，如将表中打星花部分与没打星花的部分分成两部分，则这两部分就恰好相差负号。这在部分因子设计中很重要，它构成部分因子设计表可从全因子设计表分块而得的基本思想。并且，有人还建议，如果想用序贯试验来进行试验设计，就可先用定义关系 $I=AZMPD$ 来进行第一批试验，然后，

根据试验结果看有没有必要，再用 $I=-AZMPD$ 来进行下一批试验，这样就可以做到既能有效地利用部分试验优点，在必要时，又可得到全因子试验设计所需的全部有用信息[1]。

如前所述，当因子数很多时，半因子设计的试验次数仍然不少，对子在此讨论的磷酸酶活性的实验，实际上还可加上一个温度因素（T），这样，对于全因子设计就需要做 64 次试验了，即使使用半因子设计仍有 32 次试验数，所以，统计学家就提出了一种四分之一部分因子设计法，在统计学文献中也称为 2^{6-2}部分因子设计法，从而使试验数仍保持为 16 次。从前述的分析中，可以理解，对于四分之一部分因子设计法，除去原来半因子设计已有的同名效应外，还会产生新的同名效应，即存在四个同名效应。下面将介绍四分之一部分因子设计法产生方法。首先，仍用半因子设计的产生子 $A=ZMPD$，然后再加上一个与温度（T）有关的新的产生子，即 $T=ZMP$。这样，对于四分之一部分因子设计，这样就得到了两个定义关系，即

$$I=ZMPDA \tag{4-6a}$$

和

$$I=ZMPT \tag{4-6b}$$

第三个定义关系可通过式（4-6a）和式（4-6b）两个定义关系相乘而得

$$I=ZMPDA\times ZMPT=Z^2M^2P^2DAT=DAT \tag{4-7}$$

通过上述三个定义关系，就可以算出所有的同名效应来，如要求 ZM 的同名效应时，就可以用下述三个式子计算而得

$$ZM=ZM\times ZMPDA=PDA \tag{4-8a}$$

$$ZM=ZM\times ZMPT=PT \tag{4-8b}$$

$$ZM=ZM\times DAT=ZMDAT \tag{4-8c}$$

由此算得的四分之一部分因子设计的所有同名结果列于表 4-12。

表 4-12　四分之一部分因子设计的同名效应表

同名效应						
总平均	=	ZMPDA	=	ZMPT	=	DAT
Z	=	MPDA	=	MPT	=	ZDAT
M	=	ZPDA	=	ZPT	=	MDAT
P	=	ZMDA	=	ZMT	=	MDAT
D	=	ZMPA	=	ZMPDT	=	AT
A	=	ZMPD	=	ZMPAT	=	DT
ZM	=	PDA	=	PT	=	ZMDAT
ZP	=	MDA	=	MT	=	ZPDAT
ZD	=	MPA	=	MPDT	=	ZAT
ZA	=	MPD	=	MPAT	=	ZDT
MP	=	ZDA	=	ZT	=	MPDAT
MD	=	ZPA	=	ZPDT	=	MAT
MA	=	ZPD	=	ZPAT	=	MDT
PD	=	ZMA	=	ZMDT	=	PAT
PA	=	ZMD	=	ZMAT	=	PDT
DA	=	ZMP	=	ZMPDAT	=	T

就像半因子设计一样，四分之一部分因子设计也可通过对全因子设计表进行分块后来得到不同的但却具有同样分辨率的部分因子设计表，而且，将它们合在一起就成为了六因素全因子设计表，故也可以用于进行序贯试验。它们具体可由下述不同的定义关系来构成：

$$I=ZMPDA=ZMPT=DAT \tag{4-9a}$$

$$I=-ZMPDA=-ZMPT=DAT \tag{4-9b}$$

$$I=ZMPDA=-ZMPT=-DAT \quad (4\text{-}9c)$$

$$I=-ZMPDA=ZMPT=-DAT \quad (4\text{-}9d)$$

由此可见，第一个［式(4-9a)］是原来的四分之一部分因子设计，而其余的三个［式(4-9b)～式(4-9d)］则是由对第一个设计的三个定义关系中，任意改变两个定义关系的符号而得。

三、因子设计分辨率

因子设计的分辨率是因子试验设计中的一个很重要的概念，一般说来，分辨率越高的部分因子设计越好。对于我们在上一节讨论的 2^{5-1}部分因子设计（即其定义关系为 $I=ZMPDA$ 的那个设计），其分辨率为 5；而对于定义关系为 $I=AZMP$ 的 2^{5-1}部分因于设计，其分辨率为 4。从这里我们可以看到，定义关系由几个因素组成，那么，该设计的分辨率就是几。值得提出的是，对于定义关系为 $I=ZMPDA$ 的半因子设计，它的主效应主要与四项交叉效应共现，二项交叉效应主要与三项交叉效应共现，但对于定义关系为 $I=ZMPD$ 的半因子设计，它的主效应将主要与三项交叉效应共现，二项交叉效应则与四项交叉效应共现，这样将可能丢失一些重要的三项交叉效应的信息。因此，一般可以认为，三项交叉效应要比四项交叉效应更重要，所以，分辨率为 5 的部分因于设计要比分辨率为 4 的部分因子设计好。

因子设计分辨率的正式定义为：设计的分辨率为 R 的部分因子设计为，在这样的因子设计中，无 p 项交叉效应与（$R-p$）项交叉效应共现现象的存在。值得指出的是，这一原始定义由 Boxand 给出[5]，但现在有人提出，此定义还并不完善，需要修改。有关这方面的进一步讨论，读者可参看文献［6］。部分因子设计在文献中的表述一般为数字或罗马字，例如，对于 2^{5-1}部分因子设计的分辨率可表示如下：

$$2^{5-1}(\mathrm{V});2^{5-1}(\mathrm{IV})$$

上述符号就分别表示分辨率为 5 和分辨率为 4 的 2^{5-1}部分因子设计。

对于上述给出的分辨率，一般都遵循以下三条原则：

① 一个具有分辨率为 3（$R=$ Ⅲ）的部分因子设计，其主效应之间不存在共现现象，但是，主效应一般都与二项交叉效应存在共现关系。

② 一个具有分辨率为 4（$R=$ Ⅳ）的部分因子设计，其主效应之间不存在共现现象，同时主效应与二项交叉效应也不存在共现现象，但是，主效应一般都与三项交叉效应存在共现关系。

③ 一个具有分辨率为 5（$R=$ Ⅴ）的部分因子设计，其主效应之间，主效应与二项交叉效应及三项交叉效应之间不存在共现现象，但是，主效应与四项交叉效应，二项交叉效应与三项交叉效应之间却存在着共现关系。

一般说来，构成一个高分辨的半因子设计的方法很简单，就是直接将最后一个主效应与前面所有因子的交叉效应构成一个产生于，以得到一个包括 n 个因素在内的定义关系，其余因素的主效应及各项交叉效应可由此产生。另外，一般说来，对于分辨率为 3 的部分因子设计，因为主效应一般都与二项交叉效应存在着共现现象，所以，这样的试验设计一般主要用来研究主效应，此时，这样的试验设计一般称为饱和设计。

值得提出的是，对于因子设计的研究现在仍未间断，有人提出了现代因子设计理论，可将任一因子设计中的主效应和交叉效应的计算统一在一个式子里，而且，对因子设计的分块、因子设计与最优设计的关系等都进行了详细的讨论，有兴趣的读者可参阅文献［7］。

第四节　正交试验设计和正交设计表

一、正交试验设计

正交试验设计从本质上来说，实际可以算成是一种部分因子设计方法，因其主要特征为它是基于正交表来进行试验设计和数据分析的，故有正交试验设计之称。什么是“正交表”？在定义正交表之前，有必要先介绍“完全对”的概念。

设有两组元素 a_1，a_2，…，a_n，与 b_1，b_2，…，b_m，则有 $n\times m$ 个“元素对”

$$
\begin{array}{l}
(a_1,b_1),(a_1,b_2),\cdots,(a_1,b_m)\\
(a_2,b_1),(a_2,b_2),\cdots,(a_2,b_m)\\
(a_3,b_1),(a_3,b_2),\cdots,(a_3,b_m)\\
\vdots \qquad\qquad\qquad\qquad \vdots\\
(a_n,b_1),(a_n,b_2),\cdots,(a_n,b_m)
\end{array}
$$

上述数表为两组元素构成的“完全对”。如一个矩阵的某两列中，同行元素所构成的元素对就是一个完全对，如果在此两列中，每对出现的次数也相同，则称这两列搭配均衡，否则为搭配不均衡。例如，下述矩阵的两列搭配是均衡的：

$$
\begin{array}{cc}
1 & 1\\
1 & 1\\
1 & 2\\
1 & 2\\
2 & 1\\
2 & 1\\
2 & 2\\
2 & 2
\end{array}
$$

而下述矩阵的两列则搭配不均衡：

$$
\begin{array}{cc}
1 & 1\\
1 & 2\\
2 & 1\\
2 & 2\\
1 & 2\\
1 & 2\\
2 & 2\\
2 & 2
\end{array}
$$

对于一个（$n\times m$）阶矩阵 A，它的第 j 列的元素由数码 1，2，…，t_j（$j=1$，2，…，m）所构成，如该矩阵的任意两列都搭配均衡，则称矩阵 A 为一个正交表。这里称矩阵为表，因其可将其写成表格的形式，常简记为

$$
L_n(t_1\times t_2\times t_3\times\cdots\times t_m) \tag{4-10}
$$

L 是正交表的代号，来源于拉丁方（Latin square）这一试验设计表的第一个字母，n 表示共作 n 次试验，而 $t_j(j=1,2,\cdots,m)$则代表第 j 列由 t_j 个水平组成。如所有 t_j 均相等，则可记为 $L_n(t^m)$，称为 t 水平正交表。如两列水平数不相等时，则称为混合型正交表。用正交表来安排试

验，可以达到以很小的试验次数来最大限度地获取有关各因素主效应及交叉效应的信息。有关正交试验的理论分析，读者可参阅文献[8]。正交试验的理论分析可以说明，从各种角度与标准衡量，正交设计都是一种最优设计。下面主要对如何使用正交设计表作出具体说明。

二、正交表及其交互效应表

对于每一张正交表，都有一张两列间的交互效应表与之对应，如 $L_8(2^7)$ 的两列间交互效应表为表 4-13。

表 4-13 $L_8(2^7)$的两列间交互效应表

列号	1	2	3	4	5	6	7
	(1)	3	2	5	4	7	6
		(2)	1	6	7	4	5
			(3)	7	6	5	4
				(4)	1	2	3
					(5)	3	2
						(6)	1
							(7)

从表中可查出第一列（1）与第 2 列（由(1)所在行从左向右横向找到 2）的交互效应是第 3 列，与第 4 列的交互效应是第 5 列，……，如此等等，可在表中找出任一列的交互效应列来。下面以一个简单的例子来加以说明怎样来用正交表进行试验设计和数据分析。

设有一化学反应，需考察四个实验条件的影响：① 反应温度（A）；② 反应时间（B）；③ 反应物配比（C）；④ 反应压力（D），并设各条件均是两水平。A 的两个水平为：60℃与80℃；B 的两个水平为：2.5h 与 3.5h；C 的两个水平为：1.1/1 和 1.2/1；D 的两个水平为：6.7×10^4Pa 和 8.0×10^4Pa。如想只做 8 次试验，故可选择 $L_8(2^7)$表（参见表 4-14）来安排试验。从表中可以看出，正交表的每两列都是搭配均衡的。因这个试验主要想考察 A（反应温度）与 B（反应时间）、A（反应温度）与 C（反应物配比）及 B（反应时间）与 C（反应物配比）的交叉效应，根据 $L_8(2^7)$表和其交互效应表（参见表 4-13），可将 A 因素定在 $L_8(2^7)$表的第一列，B 因素定在 $L_8(2^7)$表的第二列。从 $L_8(2^7)$的交互效应表知，第 1 列与第 2 列的交叉效应在第 3 列，将其空出，将 C 因素定在第 4 列，再由交互效应表知，第 1 列与第 4 列的交互效应在第 5 列，故也需将第 5 列空出，在根据交互效应表找出 B 因素与 C 因素的交互效应列，即第 2 列与第 4 列的交互效应列，从交互效应表可找出它们的交互效应列是第 6 列，也将其空出。于是，D 因素就定在第 7 列了。

表 4-14 $L_8(2^7)$正交试验表①

试验号 \ 列号	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

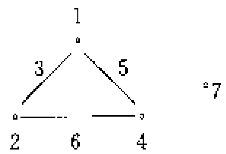
① 在此表中将上一节所用的负号用 1 来代替了，而正号则用 2 来代替了，这样可容易将正交表推广至多水平。

按这样定下各因素列号后，就可根据 $L_8(2^7)$ 正交表来安排试验了。根据试验得到的结果，然后对各交叉项效应进行估价。因为在 $L_8(2^7)$ 正交表中就可得到各交叉项效应的符号（1 为正，2 为负），采用上节所述方法，容易估价出我们所感兴趣的交叉项效应来，同样，还可以用上节所述的效应正态分布图来对它们进行显著性估价。

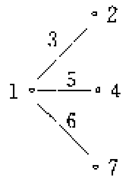
三、正交设计表的线性图及应用

对于正交设计的试验安排，除交互效应表外，还有一个很重要的工具是正交设计表的线性图，这一节我们将对此作出较详细的介绍。

对于不同的正交表，都存在其相应的标准线性图。对 $L_8(2^7)$ 正交表来说，其标准线性图如图 4-9 所示。下面以此来对正交表的线性图的使用加以说明。



(a)



(b)

图 4-9 $L_8(2^7)$ 正交表的标准线性图

由图 4-9 所示的 $L_8(2^7)$ 正交表的两个标准线性图包括以下信息：

线性图（a）：图中的点及其相应的数码代表 $L_8(2^7)$ 正交表的列，图中的连线及其相应的数码也代表 $L_8(2^7)$ 正交表中的某一列，只是这样的列是表示其线条相连的两个点所代表的两列的交互效应。从图 4-9(*a*) 中看到，该图有 4 个顶点，即点 1、2、4、7，其中 1、2、4 点有线相连，而 7 点为一孤立点。这就说明，在 $L_8(2^7)$ 正交表中，第 1 列与第2 列的交互效应列是第 3 列，第 1 列与第 4 列的交互效应列是第 5 列，第 2 列与第 4 列的交互效应列是第 6 列。第七点孤立，即说明在 $L_8(2^7)$ 正交表中不存在与第 7 列有交互效应的列。从上述讨论可知，线性图的顶点所对应的列，可用于安排试验的各个因素，如对于上节所讨论的一个四因素试验，就可将它们安排在顶点 1、2、4、7 所对应的 $L_8(2^7)$ 正交表的列之上，对那些需要调查交互效应的因素要尽量安排在顶点 1、2、4 之上，这与上节所介绍的方法是一致的。

线性图（b）：图中顶点和连线的实际意义同线性图（a），顶点为 1、2、4、7，另一不同之点是，在这种试验设计的方案中，是以 $L_8(2^7)$ 正交表中的第 1 列为中心，交叉效应将只考虑第 1 顶点与第 2、第 4 和第 7 顶点之间的，即 $L_8(2^7)$ 正交表中的第 1 列与第 2 列、第 1 列与第 4 列、第 1 列与第 7 列的交叉效应。所以，在安排试验时，将各个因素安排在顶点 1、2、4、7 之上，而且，第 1 点的选择特别重要。

上述两种正交设计的线性图都可从交互效应表中导出，对照交互效应表，都可找到它们的可能组合。所以，正交表的线性图与交互效应表是一致的，只不过线性图使用起来更为方便罢了。

四、常用正交表及相应的交互效应表与线性图

表 4-15 给出了一些常用的正交表、相应交互效应表和线性图 (Common used Orthogonal Arrays, their interaction tables and their linear graphs)，以备读者查用。对给出这些表格，在此作出几点相应说明。

① 正交表常记为

$$L_n(t_1\times t_2\times t_3\times\cdots\times t_m)$$

L 是正交表的代号，来源于拉丁方（Latin square）这一试验设计表的第一个字母；n 表示共作 n 次试验，而 t_j（$j=1$，2，…，m）则代表第 j 列由 t_j 个水平组成。如所有 t_j 均相等，则可 记为 $L_n(t^m)$，称为 t 水平正交表。如两列水平数不相等，则称为混合型正交表，记为 $L_n(t_1^m\times t_2^r)$．如 $L_{32}(2^1\times 4^9)$就表示试验次数为 32，其中一列的水平数为 2，另 9 列的水平数为 4 的正交设计表。

② 利用交互效应表，可方便进行交互效应估计。

③ 在正交表中，不同的列可分成不同的组，以方便运用线性图进行交互效应估计，根据不同组的列，运用线性图即可方便组织试验。

表 4-15　正交试验表汇集

(1) $L_4(2^3)$正交试验表①

试验号＼列号	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1
	♣	♦	♦

① ♣：第一组；♦：第二组。

L₄ 线性图

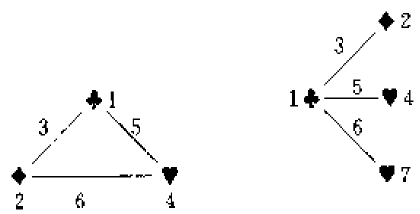


(2) $L_8(2^7)$正交试验表①

试验号＼列号	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2
	♣	♦	♦	♥	♥	♥	♥

① ♣：第一组；♦：第二组；♥：第三组。

L₈ 线性图



(1)　　(2)

(2a) $L_8(2^7)$的两列间交互效应表

列　号	1	2	3	4	5	6	7
	(1)	3	2	5	4	7	6
		(2)	1	6	7	4	5
			(3)	7	6	5	4
				(4)	1	2	3
					(5)	3	2
						(6)	1
							(7)

(3) $L_{12}(2^{11})$正交试验表[①]

试验号 \ 列号	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	2	2	2	2	2	2
3	1	1	2	2	2	1	1	1	2	2	2
4	1	2	1	2	2	1	2	2	1	1	2
5	1	2	2	1	2	2	1	2	1	2	1
6	1	2	2	2	1	2	2	1	2	1	1
7	2	1	2	2	1	1	2	2	1	2	1
8	2	1	2	1	2	2	2	1	1	1	2
9	2	1	1	2	2	2	1	2	2	1	1
10	2	2	2	1	1	1	1	2	2	1	2
11	2	2	1	2	1	2	1	1	1	2	2
12	2	2	1	1	2	1	2	1	2	2	1
	♣	♦	♦	♦	♦	♦	♦	♦	♦	♦	♦

① ♣：第一组；♦：第二组。

因为此正交表的第二组10列与交叉项"混杂"(confounded)，所以，此表最好不要用于有交叉项效应的试验设计。

(4) $L_{16}(2^{15})$正交试验表[①]

试验号 \ 列号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
3	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2
4	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
5	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
6	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1
7	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1
8	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2
9	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
10	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1
11	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1
12	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2
13	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1
14	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2
15	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2
16	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1
	♣	♦	♦	♥	♥	♥	♥	♠	♠	♠	♠	♠	♠	♠	♠

① ♣：第一组；♦：第二组；♥：第三组；♠：第四组。

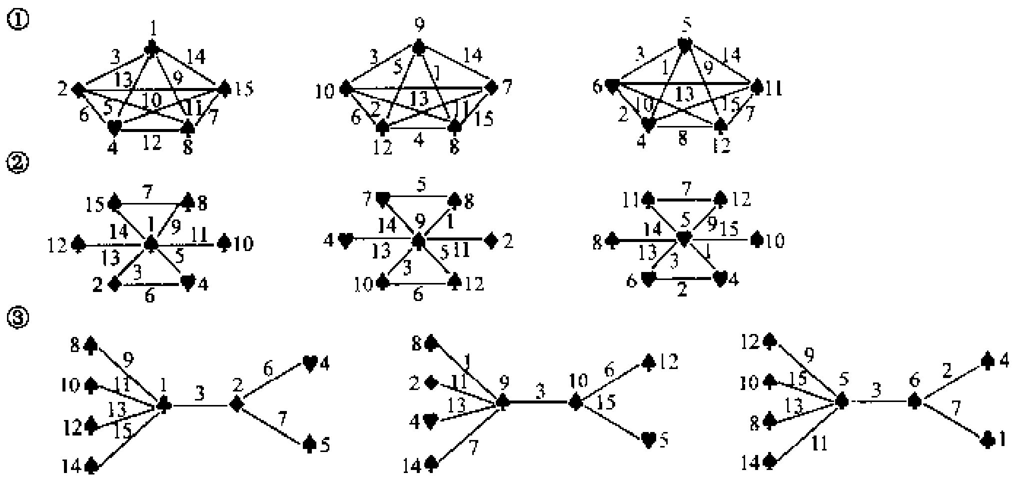
(4a) $L_{16}(2^{15})$的两列间交互效应表

列号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	(1)	3	2	5	4	7	6	9	8	11	10	13	12	15	14
		(2)	1	6	7	4	5	10	11	8	9	14	15	12	13
			(3)	7	6	5	4	11	10	9	8	15	14	13	12
				(4)	1	2	3	12	13	14	15	8	9	10	11
					(5)	3	2	13	12	15	14	9	8	11	10
						(6)	1	14	15	12	13	10	11	8	9

续表

列号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
							(7)	15	14	13	12	11	10	9	8
								(8)	1	2	3	4	5	6	7
									(9)	3	2	5	4	7	6
										(10)	1	6	7	4	5
											(11)	7	6	5	4
												(12)	1	2	3
													(13)	3	2
														(14)	1

L_{16}部分线性图

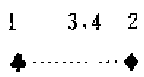


(5) $L_9(3^4)$正交试验表①

试验号 \ 列号	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1
	♣	♦	♦	♦

① ♣：第一组；♦：第二组。

L_9线性图



(6) $L_{18}(2^1\times3^7)$正交试验表①

试验号 \ 列号	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3
4	1	2	1	1	2	2	3	3
5	1	2	2	2	3	3	1	1
6	1	2	3	3	1	1	2	2
7	1	3	1	2	1	3	2	3
8	1	3	2	3	2	1	3	1
9	1	3	3	1	3	2	1	2
10	2	1	1	3	3	2	2	1

续表

试验号 \ 列号	1	2	3	4	5	6	7	8
11	2	1	2	1	1	3	3	2
12	2	1	3	2	2	1	1	3
13	2	2	1	2	3	1	3	2
14	2	2	2	3	1	2	1	3
15	2	2	3	1	2	3	2	1
16	2	3	1	3	2	3	1	2
17	2	3	2	1	3	1	2	3
18	2	3	3	2	1	2	3	1
	♣	♦	♥	♥	♥	♥	♥	♥

① ♣：第一组；♦：第二组；♥：第三组，同 L_{12}，因为交叉项"共现"(confounded)，此表最好不要用有交叉项效应的试验设计。

L_{18}线性图

1 ♣······ ···♦ 2

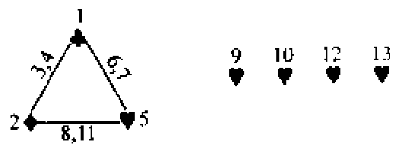
(7) $L_{27}(3^{13})$正交试验表

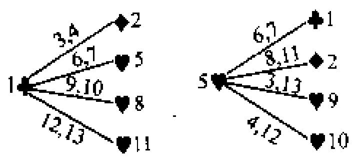
试验号 \ 列号	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3	3	3	3	3
4	1	2	2	2	1	1	1	2	2	2	3	3	3
5	1	2	2	2	2	2	2	3	3	3	1	1	1
6	1	2	2	2	3	3	3	1	1	1	2	2	2
7	1	3	3	3	1	1	1	3	3	3	2	2	2
8	1	3	3	3	2	2	2	1	1	1	3	3	3
9	1	3	3	3	3	3	3	2	2	2	1	1	1
10	2	1	2	3	1	2	3	1	2	3	1	2	3
11	2	1	2	3	2	3	1	2	3	1	2	3	1
12	2	1	2	3	3	1	2	3	1	2	3	1	2
13	2	2	3	1	1	2	3	2	3	1	3	1	2
14	2	2	3	1	2	3	1	3	1	2	1	2	3
15	2	2	3	1	3	1	2	1	2	3	2	3	1
16	2	3	1	2	1	2	3	3	1	2	2	3	1
17	2	3	1	2	2	3	1	1	2	3	3	1	2
18	2	3	1	2	3	1	2	2	3	1	1	2	3
19	3	1	3	2	1	3	2	1	3	2	1	3	2
20	3	1	3	2	2	1	3	2	1	3	2	1	3
21	3	1	3	2	3	2	1	3	2	1	3	2	1
22	3	2	1	3	1	3	2	2	1	3	3	2	1
23	3	2	1	3	2	1	3	3	2	1	1	3	2
24	3	2	1	3	3	2	1	1	3	2	2	1	3
25	3	3	2	1	1	3	2	3	2	1	2	1	3
26	3	3	2	1	2	1	3	1	3	2	3	2	1
27	3	3	2	1	3	2	1	2	1	3	1	3	2
	♣	♦	♦	♦	♥	♥	♥	♥	♥	♥	♥	♥	♥

(7a) $L_{27}(3^{13})$的两列间交互效应表

列 号	1	2	3	4	5	6	7	8	9	10	11	12	13
	(1)	3	2	2	6	5	5	9	8	8	12	11	11
		4	4	3	7	7	6	10	10	9	13	13	12
		(2)	1	1	8	9	10	5	6	7	5	6	7
			4	3	11	12	13	11	12	13	8	9	10
			(3)	1	9	10	8	7	5	6	6	7	5
				2	13	11	12	12	13	11	10	8	9
				(4)	10	8	9	6	7	5	7	5	6
					12	13	11	13	11	12	9	10	8
					(5)	1	1	2	3	4	2	4	3
						7	6	11	13	12	8	10	9
						(6)	1	4	2	3	3	2	4
							5	13	12	11	10	9	8
							(7)	3	4	2	4	3	2
								12	11	13	9	8	10
								(8)	1	1	2	3	4
									10	9	5	7	6
									(9)	1	4	2	3
										8	7	6	5
										(10)	3	4	2
											6	5	7
											(11)	1	1
												13	12
												(12)	1
													11

L_{27}部分线性图

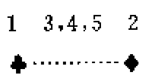




(8) $L_{16}(4^5)$正交试验表

试验号 \ 列号	1	2	3	4	5	试验号 \ 列号	1	2	3	4	5
1	1	1	1	1	1	9	3	1	3	4	2
2	1	2	2	2	2	10	3	2	4	3	1
3	1	3	3	3	3	11	3	3	1	2	4
4	1	4	4	4	4	12	3	4	2	1	3
5	2	1	2	3	4	13	4	1	4	2	3
6	2	2	1	4	3	14	4	2	3	1	4
7	2	3	4	1	2	15	4	3	2	4	1
8	2	4	3	2	1	16	4	4	1	3	2
							♣	♦	♦	♦	♦

L_{16}线性图



(9) L_{32} ($2^1 \times 4^9$) 正交试验表①

试验号 \ 列号	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3	3	3
4	1	1	4	4	4	4	4	4	4	4
5	1	2	1	1	2	2	3	3	4	4
6	1	2	2	2	1	1	4	4	3	3
7	1	2	3	3	4	4	1	1	2	2
8	1	2	4	4	3	3	2	2	1	1
9	1	3	1	2	3	4	1	2	3	4
10	1	3	2	1	4	3	2	1	4	3
11	1	3	3	4	1	2	3	4	1	2
12	1	3	4	3	2	1	4	3	2	1
13	1	4	1	2	4	3	3	4	2	1
14	1	4	2	1	3	4	4	3	1	2
15	1	4	3	4	2	1	1	2	4	3
16	1	4	4	3	1	2	2	1	3	4
17	2	1	1	4	1	4	2	3	2	3
18	2	1	2	3	2	3	1	4	1	4
19	2	1	3	2	3	2	4	1	4	1
20	2	1	4	1	4	1	3	2	3	2
21	2	2	1	4	2	3	4	1	3	2
22	2	2	2	3	1	4	3	2	4	1
23	2	2	3	2	4	1	2	3	1	4
24	2	2	4	1	3	2	1	4	2	3
25	2	3	1	3	3	1	2	4	4	2
26	2	3	2	4	4	2	1	3	3	1
27	2	3	3	1	3	1	4	2	2	4
28	2	3	4	2	2	4	3	1	3	1
29	2	4	1	3	4	2	4	2	1	3
30	2	4	2	4	3	1	3	1	2	4
31	2	4	3	1	2	4	2	4	3	1
32	2	4	4	2	1	3	1	3	4	2
	♣	♦	♥	♥	♥	♥	♥	♥	♥	♥

① ♣：第一组；♦：第二组；♥：第三组。同 L_{12}，因为交叉项“共现”(confounded)，此表最好不要用有交叉项效应的试验设计。

L_{32}线性图



(10) L_{64}(4^{21})正交试验表

试验号 \ 列号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	1	2	2	2	2	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4
6	1	2	2	2	2	2	2	2	2	1	1	1	1	4	4	4	4	3	3	3	3
7	1	2	2	2	2	3	3	3	3	4	4	4	4	1	1	1	1	2	2	2	2
8	1	2	2	2	2	4	4	4	4	3	3	3	3	2	2	2	2	1	1	1	1
9	1	3	3	3	3	1	1	1	1	3	3	3	3	4	4	4	4	2	2	2	2
10	1	3	3	3	3	2	2	2	2	4	4	4	4	3	3	3	3	1	1	1	1
11	1	3	3	3	3	3	3	3	3	1	1	1	1	2	2	2	2	4	4	4	4
12	1	3	3	3	3	4	4	4	4	2	2	2	2	1	1	1	1	3	3	3	3
13	1	4	4	4	4	1	1	1	1	4	4	4	4	2	2	2	2	3	3	3	3
14	1	4	4	4	4	2	2	2	2	3	3	3	3	1	1	1	1	4	4	4	4
15	1	4	4	4	4	3	3	3	3	2	2	2	2	4	4	4	4	1	1	1	1
16	1	4	4	4	4	4	4	4	4	1	1	1	1	3	3	3	3	2	2	2	2
17	2	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
18	2	1	2	3	4	2	1	4	3	2	1	4	3	2	1	4	3	2	1	4	3
19	2	1	2	3	4	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
20	2	1	2	3	4	4	3	2	1	4	3	2	1	4	3	2	1	4	3	2	1
21	2	2	1	4	3	1	2	3	4	2	1	4	3	3	4	1	2	4	3	2	1
22	2	2	1	4	3	2	1	4	3	1	2	3	4	4	3	2	1	3	4	1	2
23	2	2	1	4	3	3	4	1	2	4	3	2	1	1	2	3	4	2	1	4	3
24	2	2	1	4	3	4	3	2	1	3	4	1	2	2	1	4	3	1	2	3	4
25	2	3	4	1	2	1	2	3	4	3	4	1	2	4	3	2	1	2	1	4	3
26	2	3	4	1	2	2	1	4	3	4	3	2	1	3	4	1	2	1	2	3	4
27	2	3	4	1	2	3	4	1	2	1	2	3	4	2	1	4	3	4	3	2	1
28	2	3	4	1	2	4	3	2	1	2	1	4	3	1	2	3	4	3	4	1	2

续表

试验号＼列号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
29	2	4	3	2	1	1	2	3	4	4	3	2	1	2	1	4	3	3	4	1	2
30	2	4	3	2	1	2	1	4	3	3	4	1	2	1	2	3	4	4	3	2	1
31	2	4	3	2	1	3	4	1	2	2	1	4	3	4	3	2	1	1	2	3	4
32	2	4	3	2	1	4	3	2	1	1	2	3	4	3	4	1	2	2	1	4	3
33	3	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3	4	2
34	3	1	3	4	2	2	4	3	1	2	4	3	1	2	1	3	1	2	4	3	1
35	3	1	3	4	2	3	1	2	4	3	1	2	4	3	1	2	4	3	1	2	4
36	3	1	3	4	2	4	2	1	3	4	2	1	3	4	2	1	3	4	2	1	3
37	3	2	4	3	1	1	3	4	2	2	4	3	1	3	1	2	4	4	2	1	3
38	3	2	4	3	1	2	4	3	1	1	3	4	2	4	2	1	3	3	1	2	4
39	3	2	4	3	1	3	1	2	4	4	1	1	3	1	3	4	2	2	4	3	1
40	3	2	4	3	1	4	2	1	3	3	2	2	4	2	4	3	1	1	3	4	2
41	3	3	1	2	4	1	3	4	2	3	1	2	4	4	2	1	3	2	4	3	1
42	3	3	1	2	4	2	4	3	1	4	2	1	3	3	1	2	4	1	3	4	2
43	3	3	1	2	4	3	1	2	4	1	3	4	2	2	4	3	1	4	2	1	3
44	3	3	1	2	4	4	2	1	3	2	4	3	1	1	3	4	2	3	1	2	4
45	3	4	2	1	3	1	3	4	2	4	2	1	3	2	4	3	1	3	1	2	4
46	3	4	2	1	3	2	4	3	1	3	1	2	4	1	3	4	2	4	2	1	3
47	3	4	2	1	3	3	1	2	4	2	4	3	1	4	2	1	3	1	3	4	2
48	3	4	2	1	3	4	2	1	3	1	3	4	2	3	1	2	4	2	1	3	1
49	4	1	4	2	3	1	4	2	3	1	4	2	3	1	4	2	3	1	4	2	3
50	4	1	4	2	3	2	3	1	4	2	3	1	4	2	3	1	4	2	3	1	4
51	4	1	4	2	3	3	2	4	1	3	2	4	1	3	2	4	1	3	2	4	1
52	4	1	4	2	3	4	1	3	2	4	1	3	2	4	1	3	2	4	1	3	2
53	4	2	3	1	4	1	4	2	3	2	3	1	4	3	2	4	1	4	1	3	2
54	4	2	3	1	4	2	3	1	4	1	4	2	3	4	1	3	2	3	2	4	1
55	4	2	3	1	4	3	2	4	1	4	1	3	2	1	4	2	3	2	3	1	4
56	4	2	3	1	4	4	1	3	2	3	2	4	1	2	3	1	4	1	4	2	3
57	4	3	2	4	1	1	4	2	3	3	2	4	1	4	1	3	2	2	3	1	4
58	4	3	2	4	1	2	3	1	4	4	1	3	2	3	2	4	1	1	4	2	3
59	4	3	2	4	1	3	2	4	1	1	4	2	3	2	3	1	4	4	1	3	2
60	4	3	2	4	1	4	1	3	2	2	3	1	4	1	4	2	3	3	2	4	1
61	4	4	1	3	2	1	4	2	3	4	1	3	2	2	3	1	4	3	2	4	1
62	4	4	1	3	2	2	3	1	4	3	2	4	1	1	4	2	3	4	1	3	2
63	4	4	1	3	2	3	2	4	1	2	3	1	4	4	1	3	2	1	4	2	3
64	4	4	1	3	2	4	1	3	2	1	4	2	3	3	2	4	1	2	3	1	4
	1组	2组											3组								

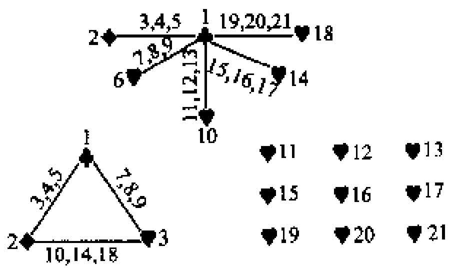
(11)$L_{64}(4^{21})$的两列间交互效应表

试验号＼列号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	(1)	3 4 5	2 4 5	2 3 5	2 3 4	7 8 9	6 8 9	6 7 9	6 7 8	11 12 13	10 12 13	10 11 13	10 11 12	15 16 17	14 16 17	14 15 17	14 15 16	19 20 21	18 20 21	18 19 21	18 19 20
		(2)	1 4 5	1 3 5	1 3 4	10 14 18	11 15 19	12 16 20	13 17 21	6 14 18	7 15 19	8 16 20	9 17 21	6 10 18	7 11 19	8 12 20	9 13 21	6 10 14	7 11 15	8 12 16	9 13 17
			(3)	1 2 5	1 2 4	11 16 21	10 17 20	13 14 19	12 15 18	7 17 20	6 16 21	9 15 18	8 14 19	8 13 19	9 12 18	6 11 21	7 10 20	9 12 15	8 13 14	7 10 17	6 11 16
				(4)	1 2 3	12 17 19	13 16 18	10 15 21	11 14 20	8 15 21	9 14 20	6 17 19	7 16 18	9 11 20	8 10 21	7 13 18	6 12 19	7 13 16	6 12 17	9 11 14	8 10 15
					(5)	13 15 20	12 14 21	11 17 18	10 16 19	9 16 19	8 17 18	7 14 21	6 15 20	7 12 21	6 13 20	9 10 19	8 11 18	8 11 17	9 10 16	6 13 15	7 12 14
						(6)	1 8 9	1 7 9	1 7 8	2 14 18	3 16 21	4 17 19	5 15 20	2 10 18	5 13 20	3 11 21	4 12 19	2 10 14	4 12 17	5 13 15	3 11 16

续表

列号 试验号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
							(7)	1 6 9	1 6 8	3 17 20	2 15 19	5 14 21	4 16 18	5 12 21	2 11 19	4 13 18	3 10 20	4 13 16	2 11 15	3 10 17	5 12 14
								(8)	1 6 7	4 15 11	5 17 18	2 16 20	3 14 19	3 13 19	4 10 21	2 12 20	5 11 18	5 11 17	3 13 14	2 12 16	4 10 15
									(9)	5 16 19	4 14 20	3 15 18	2 17 21	4 11 20	3 12 18	5 10 19	2 13 21	3 12 15	5 10 16	4 11 14	2 13 17
										(10)	1 12 13	1 11 13	1 11 12	2 6 18	4 8 21	5 9 19	3 7 20	2 6 14	5 9 16	3 7 17	4 8 15
											(11)	1 10 13	1 10 12	4 9 20	2 7 19	3 6 21	5 8 18	5 8 17	2 7 15	4 9 14	3 6 16
												(12)	1 10 11	5 7 21	3 9 18	2 8 20	4 6 19	3 9 15	4 6 17	2 8 16	5 7 14
													(13)	3 8 19	5 6 20	4 7 18	2 9 21	4 7 16	3 8 14	5 6 15	2 9 17
														(14)	1 16 17	1 15 17	1 15 16	2 6 10	3 8 13	4 9 11	5 7 12
															(15)	1 14 17	1 14 16	3 9 12	2 7 11	5 6 13	4 8 10
																(16)	1 14 15	4 7 13	5 9 10	2 8 12	3 6 11
																	(17)	5 8 11	4 6 12	3 7 10	2 9 13
																		(18)	1 20 21	1 19 21	1 19 20
																			(19)	1 18 21	1 18 20
																				(20)	1 18 19

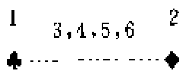
$L_{64}(4^{21})$线性图



(12) $L_{25}(5^6)$正交试验表

试验号 \ 列号	1	2	3	4	5	6	试验号 \ 列号	1	2	3	4	5	6
1	1	1	1	1	1	1	14	3	4	1	3	5	2
2	1	2	2	2	2	2	15	3	5	2	4	1	3
3	1	3	3	3	3	3	16	4	1	4	2	5	3
4	1	4	4	4	4	4	17	4	2	5	3	1	4
5	1	5	5	5	5	5	18	4	3	1	4	2	5
6	2	1	2	3	4	5	19	4	4	2	5	3	1
7	2	2	3	4	5	1	20	4	5	3	1	4	2
8	2	3	4	5	1	2	21	5	1	5	4	3	2
9	2	4	5	1	2	3	22	5	2	1	5	4	3
10	2	5	1	2	3	4	23	5	3	2	1	5	4
11	3	1	3	5	2	4	24	5	4	3	2	1	5
12	3	2	4	1	3	5	25	5	5	4	3	2	1
13	3	3	5	2	4	1		♣	♦	♦	♦	♦	♦

线性图



第五节　均匀试验设计及均匀设计表

部分因子设计和正交设计均可显著地减少实验次数又不丢失实验信息，是很好的实验设计方法，几十年来在工业生产和科学研究中得到了相当广泛的应用，其中特别是正交设计，由于正交表还具有搭配均衡的性质，故还有所谓的“均匀分散，整齐可比”的优点。然而，每一种方法都有其局限性，部分因子设计和正交设计也不例外，它们只宜用于水平数不多的实验中，若在一项试验中有 s 个因素，每个因素各有 q 个水平，用正交试验安排试验，则至少要作 q^2 次试验。例如，如果我们取 10 个水平，则需至少做 100 次试验！值得提出的是，在化学与化工的生产与科研的实际中，很多变量，如温度、压力、浓度等都是连续变量，水平数取太少有时很难反映实际情况，似乎有另辟新径的必要。

所有的实验设计方法本质上就是在试验的范围内给出挑选代表点的方法，均匀设计就是一种着重在试验范围内考虑试验点均匀散布以获得最多试验信息的较新的试验设计方法[9]，其对于具有较多水平的试验安排尤为有用。均匀设计由我国学者方开泰、王元提出，现已得到广泛应用。

均匀设计与正交设计相似，也是通过一套精心设计的表来进行实验设计的，对于每一个均匀设计表，也同时给出了一个使用表，它可指示我们如何从均匀设计表中选用适当的列来安排试验，以便使读者对均匀设计的基本思路和基本原理有一个整体概念。

一、均匀设计表的构造

均匀设计表定义：每一个均匀设计表是一个矩阵，该矩阵有 n 行 m 列，每一列是 $\{1, 2, \cdots, n\}$ 的一个置换（即 $1, 2, \cdots, n$ 的重新排列），表的第一行是 $\{1, 2, \cdots, n\}$ 的一个子集，但不一定是真子集。

与正交试验表类似，均匀设计表一般都用 $U_n(q^s)$来表示，其中 n 表示试验的次数，s 表示试验的因素数目，q 表示试验的水平数。表 4-16 给出了一个 7 次试验，7 水平 4 因素的均匀实验设计表。

表 4-16 $U_7(7^4)$均匀设计表

列号 试验号	1	2	3	4	列号 试验号	1	2	3	4
1	1	2	3	6	5	5	3	1	2
2	2	4	6	5	6	6	5	4	1
3	3	6	2	4	7	7	7	7	7
4	4	1	5	3					

从表 4-16 可以看出，对于 7 水平的实验设计，只需安排 7 次试验就够了。但符合上述定义的均匀设计表很多，在此仅介绍好格子点法构造的均匀设计表，其方法如下：

① 给定试验数 n，寻找比 n 小的整数 h，且使 n 和 h 的最大公约数为 1（即互质），符合这些条件的正整数组成一个向量 $\boldsymbol{h}=(h_1,h_2,\cdots,h_m)$。

② 均匀设计表的第 j 列生成　由下述步骤生成：

$$u_{ij}=ih_j\ [\bmod n]$$

式中，[mod n] 表示同余运算（即对 n 取余数）。u_{ij}可以递推来生成

$$u_{1j}=h_j$$

$$u_{i+1,j}=\begin{cases}u_{ij}+h_j & (u_{ij}+h_j\leqslant n)\\ u_{ij}+h_j-n & (u_{ij}+h_j>n)\end{cases}$$

$$i=1,\ 2,\ \cdots,\ n-1$$

例如，当 $n=9$ 时，符合条件①的 h 有 1、2、4、5、7、8，而 $h=3$ 或 $h=6$，因为都与9 不互质而不符合条件①。所以，U_9 表最多只可能有 6 列。又如当 $h_3=4$ 时，用上述递推公式来生成该列时，其结果依次如下：

$$u_{13}=4,\ u_{23}=4+4,\ u_{33}=8+4=12=3\ (\bmod 9)$$

$$u_{43}=3+4=7,\ u_{53}=7+4=11=2\ (\bmod 9),$$

$$u_{63}=2+4=6,\ u_{73}=6+4=10=1\ (\bmod 9),$$

$$u_{83}=1+4=5,\ u_{93}=5+4=9$$

其结果列于表 4-17 的第三列。

表 4-17 $U_9(9^6)$均匀设计表

列号 试验号	1	2	3	4	5	6	列号 试验号	1	2	3	4	5	6
1	1	2	4	5	7	8	6	6	3	6	3	6	3
2	2	4	8	1	5	7	7	7	5	1	8	4	2
3	3	6	3	6	3	6	8	8	7	5	4	2	1
4	4	8	7	2	1	5	9	9	9	9	9	9	9
5	5	1	2	7	8	4							

用上述步骤生成的均匀设计表记作 $U_n(q^m)$，向量 $\boldsymbol{h}$ 称为该表的生成向量，有时为了强调 $\boldsymbol{h}$ 的作用，可将 $U_n(q^m)$记成 $U_n(\boldsymbol{h})$。给定 n，相应的 $\boldsymbol{h}$ 可以像上例那样方便地求得，从而 m 也就确定。m 实际上是 n 的函数。这个函数曾由数学家欧拉研究过，称为欧拉函数，记为 $E(n)$。这个函数告诉我们均匀设计表最多可能有多少列。下面的结果来自数论：

(1) 当 n 为素数时，$E(n-1)=n-1$。所谓素数就是一个正整数，它与其他所有比它小的正整数互质，即最大公约数为 1，如 2，3，5，7，11，13，…，均为素数。

(2) 当 n 为素数幂时，即 n 可表示成 $n=p^m$，这里 p 为素数，m 为正整数，这时

$$E(n)=n\left(1-\frac{1}{p}\right) \tag{4-11}$$

例如 $n=9$，可表为 $n=3^2$，于是

$$E(9)=9\left(1-\frac{1}{3}\right)=6$$

即 U_9 至多可以有 6 列。

(3) 若 n 不属于上述两种情况，这时 n 一定可以表示为不同素数的方幂积，即

$$n=p_1^{m1}p_2^{m2}p_3^{m3}\cdots p_s^{ms} \tag{4-12}$$

这里 p_1，p_2，p_3，…，p_s 为不同的素数，$m1$，$m2$，…，ms 为正整数。这时

$$E(n)=n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_s}\right) \tag{4-13}$$

例如 $n=12$ 可表为 $n=2^2\times3$，于是

$$E(12)=12\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right)=4$$

即 U_{12} 至多可以有 4 列。

上述三种情况中，以素数情况为最好，最多可能获得 $(n-1)$ 列。而非素数情况，在上述表的结构中永远不可能有 $(n-1)$ 列。例如 $n=6=2^1\times3^1$，此时 $E(6)=6\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right)=2$，这说明，当 $n=6$ 时，用上述办法生成的均匀设计表只有 2 列，即最多只能安排两个因素，这是太少了。为此，王元、方开泰[10]建议，可将 $U_7(7^6)$表的最后一行去掉来构造 U_6。为了区别由上述方法生成的均匀设计表，他们把这些表记为 $U_n^*(n^m)$，在 U 的右上角加上一个“*”号。表 $U_6^*(6^6)$列于表 4-18，是由表 $U_7^*(7^6)$去掉最后一行而得。注意，由这样的方法构造出来的 U^* 表与 U 表之间存在如下关系和特点：

① 所有的 U_n^* 表是由 U_n 表划去最后一行而获得。

② U 表的最后一行全部由水平 n 组成。U_n^* 表的最后一行则不然，若每个因素的水平都是由低到高排列，U_n 表的最后一行所对应的试验将全部由最高水平组合，U_n^* 表则没有类似现象，比较容易安排试验。

③ 若 n 为偶数，U_n^* 表比 U_n 表有更多的列。如上面讨论的 U_6 只有两列，而 U_6^* 可以有 6 列。

④ 若 n 为奇数，则 U_n^* 表的列数通常少于 U_n 表。

⑤ U_n^* 表比 U_n 表有更好的均匀性，应优先采用，细节将在下节讨论。

⑥ 若将 U_n 表或 U_n^* 表的元素组成一个矩阵的秩最多分别为$\frac{E(n)+1}{2}$及$\frac{E(n+1)+1}{2}$。

表 4-18　$U_6^*(6^6)$均匀设计表

试验号＼列号	1	2	3	4	5	6	试验号＼列号	1	2	3	4	5	6
1	1	2	3	4	5	6	4	4	1	5	2	6	3
2	2	4	6	1	3	5	5	5	3	1	6	4	2
3	3	6	2	5	1	4	6	6	5	4	3	2	1

二、均匀性准则和使用表的产生

在均匀设计中，除给出的均匀设计表以外，还有一个使用表，根据此表，可以作出一些

选择，即可选出对应的列来表达试验中的因素。如要从均匀设计表 $U_n^*(n^m)$或 $U_n(n^m)$中选出 s 列，则可能有 C_m^s 种可能，若要从中选择一个最好的，这里必须对“好”和“坏”有明确的含义。$U_n(n^m)$是由它的生成向量 $h=(h_1,h_2,\cdots,h_m)$所唯一确定的，选择 s 列，本质上就是从中选择 s 个元素，并由它们来产生一个新的 $n\times s$ 矩阵，以构成一个新的均匀设计表 $U_n(h_1,\cdots,h_s)$，它的每一行是 s 维空间 R^s 的一个点，若这 n 个点在试验范围内均匀分布，则试验效果好，否则试验效果不好。比较两个可能的均匀设计表 U_n 的好坏等价于比较对应两组点集在试验空间的均匀性。于是，必须给出均匀性的度量。

度量均匀性的准则很多，其中偏差是使用历史最久，为公众所广泛接受的准则。下面将给出它的定义。

设 $U_n(n^m)$是一个均匀设计表，若把它的每一行看成 m 维空间的一个点，则 $U_n(n^m)$给出了 n 个试验点，这些点的坐标由$\{1,2,\cdots,n\}$组成。用线性变换将$\{1,2,\cdots,n\}$均匀地变到（0、1）之间如下：

$$i\to(2i-1)/(2n)\qquad i=1,2,\cdots,n \tag{4-14}$$

若用 q_{ki}表示 $U_n(n^m)$中的元素，则上面的变换等价于令

$$x_{ki}=(2q_{ki}-1)/(2n)\quad i=1,2,\cdots,m;\quad k=1,2,\cdots,n \tag{4-15a}$$

$$\boldsymbol{x}_k=(x_{k1},x_{k2},\cdots,x_{km})\qquad k=1,2,\cdots,n \tag{4-15b}$$

于是，n 个试验点变换成$[0,1]^m=C^m$ 中的 n 个点：$\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$ 在 C^m 的均匀性。

偏差定义　设 $\boldsymbol{x}_1$，$\boldsymbol{x}_2$，…，$\boldsymbol{x}_n$ 为 C^m 的 n 个点，任一向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in C^m$ 的，记 $v(x)=x_1\times x_2\times\cdots\times x_n$ 为矩形 $[0,\boldsymbol{x}]$ 的体积，n_x 为 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$ 中落入矩形 $[0,\boldsymbol{x}]$ 的点数，则

$$D(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)=\sup_{\boldsymbol{x}\in C^m}|n_x/n-v(\boldsymbol{x})| \tag{4-16}$$

称为点集 $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$ 在 C^m 中的偏差。

为什么偏差可以用于度量点集散布的均匀度呢？若 n 个点 $\boldsymbol{x}_1$，$\boldsymbol{x}_2$，…，$\boldsymbol{x}_n$ 在 C^m 中散布均匀，则 n_x/n 表示有多少比例的点落在矩形 $[0,\boldsymbol{x}]$ 中，它应当和矩形的体积相差不会太远。如果用统计学的语言来解释偏差，令

$$F_n(\boldsymbol{x})=\Big[\sum_{k=1}^{n}\mathrm{I}(\boldsymbol{x}_k\leqslant\boldsymbol{x})\Big]/n \tag{4-17}$$

表示 $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$ 的经验分布函数，式中，I（.）为示性函数。令 $F(\boldsymbol{x})$为 C^m 上均匀分布的分布函数，于是上述定义的偏差可表为

$$D(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)=\sup_{\boldsymbol{x}\in C^m}|F_n(\boldsymbol{x})-F(\boldsymbol{x})| \tag{4-18}$$

偏差实际上就是在分布拟合检验中的 Kolmogorov-Smirrov 统计量，它给出了经验和理论分布之间的偏差。

在 C^m 中任给 n 个点 $\boldsymbol{x}_1$，$\boldsymbol{x}_2$，…，$\boldsymbol{x}_n$，如何计算它们的偏差对均匀设计表的构造十分重要。长期以来，一直没有人提出一个实用的算法。方开泰等是把偏差展开成级数，取其首项，给出近似偏差的准则。此法计算方便，但有时误差较大，而且只适合好格子点法构造的均匀设计，不能计算正交设计等其他方法所产生的试验点的偏差。最近，Bundschuh and 和朱尧辰[11]给出了计算偏差的算法，当因素不太多时，该算法可精确地求出任何点集的偏差。

设要从均匀设计表 $U_n(n^m)$中选出 s 列，使其相应的均匀设计有最小偏差。当 m 和 s 较大

时，由 m 列中取出 s 列的数目有 C_m^s 之多，要比较这么多组点集的均匀性工作量很大，需用简化计算和近似求解的方法。这里仅介绍利用整数的同余幂来产生 h_1，h_2，…，h_s 的方法。

令 a 为小于 n 的整数，且 a，a^2 (mod n)，…，a^t (mod n) 互不相同，$a^{t+1}=1$ (mod n)，则称 a 对 n 的次数为 t，例如

$$2^1=2,\ 2^2=4,\ 2^3=3,\ 2^4=1 \pmod 5$$

则 2 对 5 的次数为 3，又如

$$3^1=3,\ 3^2=9,\ 3^3=5,\ 3^4=4,\ 3^5=1 \pmod{11}$$

则 3 对 11 的次数为 4。一般若 a 对 n 的次数大于或等于 $s-1$，且 a 与 n 互质，则可用

$$(a^0,\ \cdots,\ a^{s-1}) \pmod n$$

作为生成向量，故 a 称为均匀设计的生成元。然后在一切可能的 a（最多 $n-1$ 个）中去比较相应试验点的均匀性，工作量则可大大减少。理论和实践证明，这种方法获得的均匀设计使用表仍能保证设计的均匀性。于是对给定的 n 和 s，只要求得最优的 a，便可获得生成向量，从而获得相应的均匀设计表。

表 4-19　U_n 和 U_n^* 的生成元和相应设计的偏差

n	2	3	4	5	6	7
5	2(0.3100)	2(0.4570)				
6	3(0.1875)	3(0.2656)	3(0.2990)			
7	3(0.2398)	3(0.3721)	3(0.4760)			
8	4(0.1445)	4(0.2000)	2(0.2709)			
9	4(0.1944)	4(0.3102)	2(0.4066)			
10	7(0.1125)	7(0.1681)	5(0.2236)	5(0.2411)	7(0.2994)	
11	7(0.1632)	7(0.2649)	5(0.3528)	7(0.4286)	7(0.4942)	
12	5(0.1163)	6(0.1838)	6(0.2233)	4(0.2272)	6(0.2670)	6(0.2768)
13	5(0.1405)	6(0.2308)	6(0.3107)	4(0.3814)	6(0.4439)	6(0.4992)
14	11(0.0957)	7(0.1455)	7(0.2091)			
15	11(0.1233)	7(0.2043)	7(0.2772)			
16	10(0.0908)	5(0.1262)	5(0.1705)	5(0.2070)	10(0.2518)	2(0.2769)
17	11(0.1099)	10(0.1832)	10(0.2501)	10(0.3111)	10(0.3667)	10(0.4174)
18	8(0.0779)	9(0.1394)	9(0.1754)	4(0.2047)	3(0.2245)	9(0.2247)
19	8(0.0990)	8(0.1660)	14(0.2277)	14(0.2845)	14(0.3368)	14(0.3850)
20	13(0.0947)	5(0.1363)	10(0.1915)	10(0.2012)	10(0.2010)	
21	13(0.0947)	10(0.1581)	10(0.2089)	10(0.2620)	10(0.3113)	
22	9(0.0677)	17(0.1180)	17(0.1392)	17(0.1827)	17(0.1930)	11(0.2195)
23	17(0.0827)	15(0.1397)	17(0.1930)	11(0.2428)	17(0.2893)	11(0.3328)
24	11(0.0586)	6(0.1031)	6(0.1441)	12(0.1758)	12(0.2064)	12(0.2198)
25	11(0.0764)	11(0.1294)	11(0.1793)	11(0.2261)	4(0.2701)	9(0.3115)
26	16(0.0588)	10(0.1136)	5(0.1311)	5(0.1683)	16(0.1828)	5(0.1967)
27	20(0.0710)	20(0.1205)	20(0.1673)	20(0.2115)	16(0.2533)	16(0.2927)
28	18(0.0545)	7(0.0935)	7(0.1074)	16(0.1381)	7(0.1578)	7(0.1550)
29	23(0.0663)	9(0.1128)	7(0.1596)	16(0.1937)	16(0.2384)	16(0.2760)
30	22(0.0519)	22(0.0888)	18(0.1325)	18(0.1465)	18(0.1621)	11(0.1924)
31	14(0.0622)	12(0.1060)	22(0.1477)	12(0.1874)	12(0.2251)	22(0.2611)
37	17(0.0524)	23(0.0930)	17(0.1255)	7(0.1599)	7(0.1929)	7(0.2245)

表 4-20 奇数 n 的 U_n^* 表的生成矢量和相应设计的偏差

n	s	生成矢量	D	$P/\%$
7	2	(1,5)	0.1582	34.03
	3	(3,5,7)	0.2132	42.70
9	2	(1,5)	0.1574	19.03
	3	(3,7,9)	0.1980	36.17
11	2	(1,5)	0.1136	30.39
	3	(5,7,11)	0.2307	12.91
13	2	(1,9)	0.0962	31.53
	3	(1,9,11)	0.1442	37.52
	4	(1,5,9,11)	0.2076	33.18
15	2	(1,7)	0.0833	32.44
	3	(1,5,13)	0.1361	33.38
	4	(5,7,9,13)	0.1511	45.49
	5	(5,7,9,11,15)	0.2090	24.60
17	2	(1,7)	0.0856	22.11
	3	(1,7,13)	0.1331	27.35
	4	(7,11,13,17)	0.1785	28.63
19	2	(1,9)	0.0755	23.74
	3	(1,3,11)	0.1372	17.35
	4	(1,3,7,11)	0.1807	20.64
	5	(7,9,11,13,19)	0.1897	33.32
21	2	(1,13)	0.0679	28.30
	3	(1,7,9)	0.1121	29.10
	4	(1,5,7,13)	0.1381	33.89
	5	(1,9,13,17,19)	0.1759	32.86
23	2	(1,17)	0.0638	29.62
	3	(11,17,19)	0.1029	26.34
	4	(1,7,13,19)	0.1310	32.12
	5	(11,13,17,19,23)	0.1691	30.35
25	2	(1,11)	0.0588	23.04
	3	(3,5,25)	0.0975	24.65
	4	(5,7,9,25)	0.1210	32.52
	5	(11,15,17,19,21)	0.1532	32.24
27	2	(1,11)	0.0600	15.49
	3	(1,9,15)	0.1009	16.27
	4	(1,11,15,25)	0.1189	28.93
	5	(3,13,17,19,27)	0.1378	34.85
29	2	(1,19)	0.0520	16.27
	3	(1,17,19)	0.0914	18.97
	4	(1,17,19,23)	0.1050	34.21
	5	(13,17,19,23,29)	0.1730	12.93
31	2	(1,9)	0.0554	10.93
	3	(1,9,19)	0.0908	14.34
	4	(3,13,21,27)	0.1100	25.52
	5	(5,9,11,17,19)	0.1431	23.64

表 4-19 对奇数 $n(5\leqslant n\leqslant 31, n=37)$给出了 U_n 表的生成元及其均匀设计的偏差，同时对偶数 $n(6\leqslant n\leqslant 30)$给出了 U_n^* 表的生成元及相应的偏差。类似地，对奇数 n，也可获得 U_n^* 表的

生成向量及相应的偏差(表 4-20)。此两表由方开泰等算出[12]。综合两个表的结果,可作如下说明:

① 对奇数 n,U_n^* 表比 U_n 表有更好的均匀性。例如 $n=15,s=4$ 时,$U_n(15^4)$的偏差为 $D=0.2772$,而 $U_n^*(15^4)$的偏差为 $D=0.1511$,后者比前者相对降低了 45.49%。

② 若 n 固定,当 s 增大时,U_n 表(或 U_n^* 表)的偏差也随之增大,若 s 固定,U_n 表的偏差随 n 的增大而减少,而 U_n^* 表的偏差一般也随之增大而减少,但有少数例外、其原因是它们的 U_{n+1}表的可能列数 $E(n+1)$不太多,但其中选择 s 的可能组合也不多,从而最小偏差相对偏大。

③ 表 4-19 列举的 U_n 表和 U_n^* 表是由生成元方法生成的,而表 4-20 的 U_n^* 表是考虑从 U_{n+1}表中选出 s 列的一切可能的组合,所以生成向量中不一定包含 1。

均匀度度量的方法很多,最初王元、方开泰[10]提出了近似偏差的均匀性准则,利用这个准则,他们给出了 $n\leqslant 31$ 的使用表。丁元[13]利用最优实验设计理论中的 A 最优和 D 最优准则,给出了相应的使用表,蒋声和陈瑞琛[14,15]从几何的观点提出了体积距离的度量。方开泰和郑胡灵[16]也从几何的角度建议用最大对称差的条件来度量均匀性,并提出了均匀性度量必须要满足的条件。

三、拟水平均匀设计

由于在实际的化学实验和化工生产中情况变化很多,有时各个因素的水平数是各自不同的,唯如此方可与实际情况相同,所以,在实验设计中就有所谓的拟水平方法,即针对实验设计诸因素的水平数存在不相同的情况下的一种设计,有时也称为混合水平设计。先看一个简单的例子。若在一个试验中,有两个因素 A 和 B 为三水平,一个因素 C 为二水平,这个试验可用正交表 $L_{18}(2^1\times 3^7)$来安排,这等价于全面试验,并且不可能找到比 L_{18}更小的正交表来安排这个试验。直接运用均匀设计似也困难,但如引入所谓拟水平技术,就可用均匀设计表来安排试验了。对此例,可用 $U_6^*(6^6)$表中(表 4-18)的前三列来进行安排,将因素 A 和 B 放在前两列,将 C 放在第三列,继续将前两列的水平两两合并,即:$\{1,2\}\Rightarrow 1,\{3,4\}\Rightarrow 2,\{5,6\}\Rightarrow 6$;第三列则合并成为二水平:$\{1,2,3\}\Rightarrow 1,\{4,5,6\}\Rightarrow 2$,这样,可得如表 4-21 所示的拟水平均匀设计表 $U_6(3^2\times 2)$。这个表具有很好的均衡性,例如,A 列和 C 列,B 列和 C 列的二因素设计正好组成它们的全面试验方案,A 列和 B 列的两因素设计中也没有重复试验。

表 4-21 拟水平设计 $U_6(3^2\times 2)$

试验号＼列号	A	B	C	试验号＼列号	A	B	C
1	(1)1	(2)1	(3)1	4	(4)2	(1)1	(5)2
2	(2)1	(4)2	(6)2	5	(5)3	(3)2	(1)2
3	(3)2	(6)3	(2)1	6	(6)3	(5)3	(4)2

但并不是每一次作拟水平设计都能这么好。如要安排一个二因素(A,B)五水平和一因素(C)二水平的试验,若用正交设计,可选 L_{50}表,但试验次数太多,若用均匀设计,可用表 4-24(7)$U_{10}^*(10^8)$表的第 1、5、6 列来进行来安排。将 1、5 列的每两水平进行合并,即:$\{1,2\}\Rightarrow 1,\{3,4\}\Rightarrow 2,\cdots,\{9,10\}\Rightarrow 5$;而将 7 列的每五水平进行合并,即:$\{1,2,3,4,5\}\Rightarrow 1$,$\{6,7,8,9,10\}\Rightarrow 2$,于是可得到如表 4-22 所示的拟水平均匀设计。此表的 A 和 C 两列,有两个(2,2),但没有(2,1),有两个(4,1),却没有(4,2),因此均衡性不好。

表 4-22 拟水平设计 $U_{10}(5^2\times2)$ （一）

试验号＼列号	A	B	C	试验号＼列号	A	B	C
1	(1)1	(5)3	(7)2	6	(6)3	(8)4	(9)2
2	(2)1	(10)5	(3)1	7	(7)4	(2)1	(5)1
3	(3)2	(4)2	(10)2	8	(8)4	(7)4	(1)1
4	(4)2	(9)5	(6)2	9	(9)5	(1)1	(8)2
5	(5)3	(3)2	(2)1	10	(10)5	(6)3	(4)1

因 $U_{10}^*(10^8)$的均匀设计表实际有 8 列，其实，在这 8 列任取 3 列的一个组合都可以用来设计，例如，如果取用 $U_{10}^*(10^8)$表的第 1、2、5 列来进行上述拟水平安排，可得如表 4-23 的 $U_{10}(5^2\times2)$设计，注意此设计既具有均衡性，而且其偏差 $D=0.3925$，达到了 $U_{10}^*(10^8)$均匀设计表所有任取三列来组合所得偏差的最小值。所以，由表 4-23 示出的拟水平设计似应比表 4-22 示出的要好。

表 4-23 拟水平设计 $U_{10}(5^2\times2)$（二）

试验号＼列号	A	B	C	试验号＼列号	A	B	C
1	(1)1	(2)1	(5)1	6	(6)3	(1)1	(8)2
2	(2)1	(4)2	(10)2	7	(7)4	(3)2	(2)1
3	(3)2	(6)3	(4)1	8	(8)4	(5)3	(7)2
4	(4)2	(8)4	(9)2	9	(9)5	(7)4	(1)1
5	(5)3	(10)5	(3)1	10	(10)5	(9)5	(6)2

四、均匀设计表

本节将试验次数从 5 到 26 均匀设计表收集于此节（表 4-24），以供读者方便选用，注意到均匀设计的试验次数是可与水平数相对应的，也就是说，可以据此来安排具有 26 水平的试验了。因素的个数也最高可达 12 个。在此节中同时提供的使用表给出了每种可能组合的偏差值，读者可根据具体的情况，即试验次数、水平数及均衡性等条件与偏差值一起来综合考虑，灵活选用。

表 4-24 均匀设计表汇集

(1) $U_5(5^3)$均匀设计表

试验号	1	2	3	试验号	1	2	3
1	1	2	4	4	4	3	1
2	2	4	3	5	5	5	5
3	3	1	2				

(1a) $U_5(5^3)$使用表

维　数	列　号	偏　差
2	1,2	0.3100
3	1,2,3	0.4570

(2) $U_6^*(6^4)$均匀设计表

试验号	1	2	3	4	试验号	1	2	3	4
1	1	2	3	6	4	4	1	5	3
2	2	4	6	5	5	5	3	1	2
3	3	6	2	4	6	6	5	4	1

(2a) $U_6^*(6^4)$使用表

维　　数	列　　号	偏　　差
2	1,3	0.1875
3	1,2,3	0.2656
4	1,2,3,4	0.2900

(3) $U_7^*(7^4)$均匀设计表

试验号	1	2	3	4	试验号	1	2	3	4
1	1	3	5	7	5	5	7	1	3
2	2	6	2	6	6	6	2	6	2
3	3	1	7	5	7	7	5	3	1
4	4	4	4	4					

(3a) $U_7^*(7^4)$使用表

维　　数	列　　号	偏　　差
2	1,3	0.1582
3	2,3,4	0.2132

(4) $U_8^*(8^5)$均匀设计表

试验号	1	2	3	4	5	试验号	1	2	3	4	5
1	1	2	4	7	8	5	5	1	2	8	4
2	2	4	8	5	7	6	6	3	6	6	3
3	3	6	3	3	6	7	7	5	1	4	2
4	4	8	7	1	5	8	8	7	5	2	1

(4a) $U_8^*(8^5)$使用表

维　　数	列　　号	偏　　差
2	1,3	0.1445
3	1,3,4	0.2000
4	1,2,3,5	0.2709

(5) $U_9(9^5)$均匀设计表

试验号	1	2	3	4	5	试验号	1	2	3	4	5
1	1	2	4	7	8	6	6	3	6	6	3
2	2	4	8	5	7	7	7	5	1	4	2
3	3	6	3	3	6	8	8	7	5	2	1
4	4	8	7	1	5	9	9	9	9	9	9
5	5	1	2	8	4						

(5a) $U_9(9^5)$使用表

维　　数	列　　号	偏　　差
2	1,3	0.1944
3	1,3,4	0.3102
4	1,2,3,5	0.4066

(6) $U_{11}(11^6)$均匀设计表

试验号	1	2	3	4	5	6	试验号	1	2	3	4	5	6
1	1	2	3	5	7	10	7	7	3	10	2	5	4
2	2	4	6	10	5	9	8	8	5	2	7	1	3
3	3	6	9	4	10	8	9	9	7	5	1	8	2
4	4	8	1	9	6	7	10	10	9	8	6	4	1
5	5	10	4	3	2	6	11	11	11	11	11	11	11
6	6	1	7	8	9	5							

(6a) $U_{11}(11^6)$使用表

维　数	列　号	偏　差	维　数	列　号	偏　差
2	1,5	0.1632	5	1,2,3,4,5	0.4286
3	1,4,5	0.2649	6	1,2,3,4,5,6	0.4942
4	1,3,4,5	0.3528			

(7) $U_{10}^*(10^8)$均匀设计表

试验号	1	2	3	4	5	6	7	8
1	1	2	3	4	5	7	9	10
2	2	4	6	8	10	3	7	9
3	3	6	9	1	4	10	5	8
4	4	8	1	5	9	6	3	7
5	5	10	4	9	3	2	1	6
6	6	1	7	2	8	9	10	5
7	7	3	10	6	2	5	8	4
8	8	5	2	10	7	1	6	3
9	9	7	5	3	1	8	4	2
10	10	9	8	7	6	4	2	1

(7a) $U_{10}^*(10^8)$使用表

维　数	列　号	偏　差	维　数	列　号	偏　差
2	1,6	0.1125	5	1,3,4,5,7	0.2414
3	1,5,6	0.1681	6	1,2,3,5,6,8	0.2994
4	1,3,4,5	0.2236			

(8) $U_{12}^*(12^{10})$均匀设计表

试验号	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	8	9	10	12
2	2	4	6	8	10	12	3	5	7	11
3	3	6	9	12	2	5	11	1	4	10
4	4	8	12	3	7	11	6	10	1	9
5	5	10	2	7	12	4	1	6	11	8
6	6	12	5	11	4	10	9	2	8	7
7	7	1	8	2	9	3	4	11	5	6
8	8	3	11	6	1	9	12	7	2	5
9	9	5	1	10	6	2	7	3	12	4
10	10	7	4	1	11	8	2	12	9	3
11	11	9	7	5	3	1	10	8	6	2
12	12	11	10	9	8	7	5	4	3	1

(8a) $U_{12}^*(12^{10})$使用表

维　数	列　号	偏　差	维　数	列　号	偏　差
2	1,5	0.1163	5	1,3,4,8,10	0.2272
3	1,6,9	0.1838	6	1,2,6,7,8,9	0.2670
4	1,6,7,9	0.2233	7	1,2,6,7,8,9,10	0.2768

(9) $U_{13}(13^8)$均匀设计表

试验号	1	2	3	4	5	6	7	8
1	1	2	5	6	8	9	10	12
2	2	4	10	12	3	5	7	11
3	3	6	2	5	11	1	4	10
4	4	8	7	11	6	10	1	9
5	5	10	12	4	1	6	11	8
6	6	12	4	10	9	2	8	7
7	7	1	9	3	4	11	5	6
8	8	3	1	9	12	7	2	5
9	9	5	6	2	7	3	12	4
10	10	7	11	8	2	12	9	3
11	11	9	3	1	10	8	6	2
12	12	11	8	7	5	4	3	1
13	13	13	13	13	13	13	13	13

(9a) $U_{13}(13^8)$使用表

维数	列号	偏差	维数	列号	偏差
2	1,3	0.1405	5	1,4,5,6,7	0.3814
3	1,4,7	0.2308	6	1,2,4,5,6,7	0.4439
4	1,4,5,7	0.3107	7	1,2,4,5,6,7,8	0.4992

(10) $U_{14}^*(14^5)$均匀设计表

试验号	1	2	3	4	5	试验号	1	2	3	4	5
1	1	4	7	11	13	8	8	2	11	13	14
2	2	8	14	7	11	9	9	6	3	9	12
3	3	12	6	3	9	10	10	10	10	5	10
4	4	1	13	14	7	11	11	14	2	1	8
5	5	5	5	10	5	12	12	3	9	12	6
6	6	9	12	6	3	13	13	7	1	8	4
7	7	13	4	2	1	14	14	11	8	4	2

(10a) $U_{14}^*(14^5)$使用表

维数	列号	偏差	维数	列号	偏差
2	1,4	0.0957	4	1,2,3,5	0.2091
3	1,2,3	0.1455			

(11) $U_{15}^*(15^7)$均匀设计表

试验号	1	2	3	4	5	6	7	试验号	1	2	3	4	5	6	7
1	1	5	7	9	11	13	15	9	9	13	15	1	3	5	7
2	2	10	14	2	6	10	14	10	10	2	6	10	14	2	6
3	3	15	5	11	1	7	13	11	11	7	13	3	9	15	5
4	4	4	12	4	12	4	12	12	12	12	4	12	4	12	4
5	5	9	3	13	7	1	11	13	13	1	11	5	15	9	3
6	6	14	10	6	2	14	10	14	14	6	2	14	10	6	2
7	7	3	1	15	13	11	9	15	15	11	9	7	5	3	1
8	8	8	8	8	8	8	8								

(11a) $U_{15}^*(15^7)$使用表

维　　数	列　　　号	偏　　差	维　　数	列　　　号	偏　　差
2	1,3	0.0833	4	1,2,4,6	0.1551
3	1,2,6	0.1361	5	2,3,4,5,7	0.2272

(12) $U_{16}^*(16^{10})$均匀设计表

试验号	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	4	5	6	8	9	10	13	14	15	16
2	2	4	8	10	12	16	1	3	9	11	13	15
3	3	6	12	15	1	7	10	13	5	8	11	14
4	4	8	16	3	7	15	2	6	1	5	9	13
5	5	10	3	8	13	6	11	16	14	2	7	12
6	6	12	7	13	2	14	3	9	10	16	5	11
7	7	14	11	1	8	5	12	2	6	13	3	10
8	8	16	15	6	14	13	4	12	2	10	1	9
9	9	1	2	11	3	4	13	5	15	7	16	8
10	10	3	6	16	9	12	5	15	11	4	14	7
11	11	5	10	4	15	3	14	8	7	1	12	6
12	12	7	14	9	4	11	6	1	3	15	10	5
13	13	9	1	14	10	2	15	11	16	12	8	4
14	14	11	5	2	16	10	7	4	12	9	6	3
15	15	13	9	7	5	1	16	14	8	6	4	2
16	16	15	13	12	11	9	8	7	4	3	2	1

(12a) $U_{16}^*(16^{10})$使用表

维　　数	列　　　号	偏　　差	维　　数	列　　　号	偏　　差
2	1,8	0.0908	5	1,4,5,6,9	0.2070
3	1,4,6	0.1262	6	1,3,5,8,10,11	0.2518
4	1,4,5,6	0.1705	7	1,2,3,6,9,11,12	0.2769

(13) $U_{18}^*(18^{11})$均匀设计表

试　验　号	1	2	3	4	5	6	7	8	9	10	11
1	1	3	4	5	6	7	8	9	11	15	16
2	2	6	8	10	12	14	16	18	3	11	13
3	3	9	12	15	18	2	5	8	14	7	10
4	4	12	16	1	5	9	13	17	6	3	7
5	5	15	1	6	11	16	2	7	17	18	4
6	6	18	5	11	17	4	10	16	9	14	1
7	7	2	9	16	4	11	18	6	1	10	17
8	8	5	13	2	10	18	7	15	12	6	14
9	9	8	17	7	16	6	15	5	4	2	11
10	10	11	2	12	3	13	4	14	15	17	8
11	11	14	6	17	9	1	12	4	7	13	5
12	12	17	10	3	15	8	1	13	18	9	2
13	13	1	14	8	2	15	9	3	10	5	18
14	14	4	18	13	8	3	17	12	2	1	15
15	15	7	3	18	14	10	6	2	13	16	12
16	16	10	7	4	1	17	14	11	5	12	9
17	17	13	11	9	7	5	3	1	16	8	6
18	18	16	15	14	13	12	11	10	8	4	3

(13a) $U_{18}^{*}(18^{11})$使用表

维　数	列　　号	偏　差	维　数	列　　号	偏　差
2	1,7	0.0779	5	1,3,6,8,11	0.2047
3	1,4,8	0.1394	6	1,2,4,7,8,10	0.2245
4	1,4,6,8	0.1754	7	1,4,5,6,8,9,11	0.2247

(14) $U_{22}^{*}(22^{11})$均匀设计表

试验号	1	2	3	4	5	6	7	8	9	10	11
1	1	5	6	8	9	11	13	14	17	20	21
2	2	10	12	16	18	22	3	5	11	17	19
3	3	15	18	1	4	10	16	19	5	14	17
4	4	20	1	9	13	21	6	10	22	11	15
5	5	2	7	17	22	9	19	1	16	8	13
6	6	7	13	2	8	20	9	15	10	5	11
7	7	12	19	10	17	8	22	6	4	2	9
8	8	17	2	18	3	19	12	20	21	22	7
9	9	22	8	3	12	7	2	11	15	19	5
10	10	4	14	11	21	18	15	2	9	16	3
11	11	9	20	19	7	6	5	16	3	13	1
12	12	14	3	4	16	17	18	7	20	10	22
13	13	19	9	12	2	5	8	21	14	7	20
14	14	1	15	20	11	16	21	12	8	4	18
15	15	6	21	5	20	4	11	3	2	1	16
16	16	11	4	13	6	15	1	17	19	21	14
17	17	16	10	21	15	3	14	8	13	18	12
18	18	21	16	6	1	14	4	22	7	5	10
19	19	3	22	14	10	2	17	13	1	12	8
20	20	8	5	22	19	13	7	4	18	9	6
21	21	13	11	7	5	1	20	18	12	6	4
22	22	18	17	15	14	12	10	9	6	3	2

(14a) $U_{22}^{*}(22^{11})$使用表

维　数	列　　号	偏　差	维　数	列　　号	偏　差
2	1,5	0.0677	5	1,4,7,8,9	0.1827
3	1,7,9	0.1108	6	1,4,7,8,9,11	0.1930
4	1,7,8,9	0.1392	7	1,2,3,5,6,7,10	0.2195

(15) $U_{24}^{*}(24^{9})$均匀设计表

试验号	1	2	3	4	5	6	7	8	9
1	1	3	6	7	9	11	12	16	19
2	2	6	12	14	18	22	24	7	13
3	3	9	18	21	2	8	11	23	7
4	4	12	24	3	11	19	23	14	1
5	5	15	5	10	20	5	10	5	20
6	6	18	11	17	4	16	22	21	14
7	7	21	17	24	13	2	9	12	8
8	8	24	23	6	22	13	21	3	2
9	9	2	4	13	6	24	8	19	21
10	10	5	10	20	15	10	20	10	15
11	11	8	16	2	24	21	7	1	9

续表

试验号	1	2	3	4	5	6	7	8	9
12	12	11	22	9	8	7	19	17	3
13	13	14	3	16	17	18	6	8	22
14	14	17	9	23	1	4	18	24	16
15	15	20	15	5	10	15	5	15	10
16	16	23	21	12	19	1	17	6	4
17	17	1	2	19	3	12	4	22	23
18	18	4	8	1	12	23	16	13	17
19	19	7	14	8	21	9	3	4	11
20	20	10	20	15	5	20	15	20	5
21	21	13	1	22	14	6	2	11	24
22	22	16	7	4	23	17	14	2	18
23	23	19	13	11	7	3	1	18	12
24	24	22	19	18	16	14	13	9	6

(15a) $U_{24}^{*}(24^{9})$使用表

维数	列号	偏差	维数	列号	偏差
2	1,6	0.0586	5	1,2,6,7,9	0.1758
3	1,3,6	0.1031	6	1,2,4,6,7,9	0.2064
4	1,3,6,8	0.1441	7	1,2,4,5,6,7,9	0.2198

(16) $U_{26}^{*}(26^{11})$均匀设计表

试验号	1	2	3	4	5	6	7	8	9	10	11
1	1	4	5	7	10	13	16	17	19	20	25
2	2	8	10	14	20	26	5	7	11	13	23
3	3	12	15	21	3	12	21	24	3	6	21
4	4	16	20	1	13	25	10	14	22	26	19
5	5	20	25	8	23	11	26	4	14	19	17
6	6	24	3	15	6	24	15	21	6	12	15
7	7	1	8	22	16	10	4	11	25	5	13
8	8	5	13	2	26	23	20	1	17	25	11
9	9	9	18	9	9	9	9	18	9	18	9
10	10	13	23	16	19	22	25	8	1	11	7
11	11	17	1	23	2	8	14	25	20	4	5
12	12	21	6	3	12	21	3	15	12	24	3
13	13	25	11	10	22	7	19	5	4	17	1
14	14	2	16	17	5	20	8	22	23	10	26
15	15	6	21	24	15	6	24	12	15	3	24
16	16	10	26	4	25	19	13	2	7	23	22
17	17	14	4	11	8	5	2	19	26	16	20
18	18	18	9	18	18	18	18	9	18	9	18
19	19	22	14	25	1	4	7	26	10	2	16
20	20	26	19	5	11	17	23	16	2	22	14
21	21	3	24	12	21	3	12	6	21	15	12
22	22	7	2	19	4	16	1	23	13	8	10
23	23	11	7	26	14	2	17	13	5	1	8
24	24	15	12	6	24	15	6	3	24	21	6
25	25	19	17	13	7	1	22	20	16	14	4
26	26	23	22	20	17	14	11	10	8	7	2

(16a) $U_{26}^{*}(26^{11})$使用表

维　数	列　　号	偏　差	维　数	列　　号	偏　差
2	1,7	0.0588	5	1,2,3,8,11	0.1682
3	1,5,9	0.1126	6	1,2,4,6,7,9	0.1828
4	1,3,8,11	0.1311	7	1,2,3,8,9,10,11	0.1967

第六节　最优试验设计

前几节所有讨论的试验设计方法都是假定模型的设计矩阵 $\boldsymbol{X}$ 已经给定，实际上，所有试验设计的统计推断结果均与 $\boldsymbol{X}$ 的值有着密切的关系。因此，如何在试验前（若能做到的话），适当地选择因素的值（即 $\boldsymbol{X}$ 矩阵），使得 $\boldsymbol{X}$ 矩阵在统计推断中能具有某种统计优良性质，这个问题是十分重要的，也是近 20 多年来试验设计中一个活跃的研究课题，本节拟讨论的最优设计法就包括这方面的问题，希望更多了解这方面情况的读者可参看有关专著[17~19]。

一、最优设计的各种准则

为讨论问题的方便，在此将前述的试验设计回归模型改写为以下更为一般的回归模型

$$y=\beta_1 f_1(\boldsymbol{x})+\beta_2 f_2(\boldsymbol{x})+\cdots+\beta_m f_m(\boldsymbol{x})+e \tag{4-19}$$

其中 $\boldsymbol{x}$ 为试验因素空间的一点，如试验因素空间是 p 维欧氏空间，那么 $\boldsymbol{x}$ 就是 p 维向量，$f_i(\boldsymbol{x})(i=1,2,\cdots,m)$都是该试验因素空间的点的连续函数。例如，当试验因素空间为 2 维时，即 $\boldsymbol{x}=(x_1,x_2)$，如取因素的所有二次项来构成试验设计的回归模型的话，就可取

$$f_1(\boldsymbol{x})=1$$
$$f_2(\boldsymbol{x})=x_1$$
$$f_3(\boldsymbol{x})=x_2$$
$$f_4(\boldsymbol{x})=x_1x_2$$
$$f_5(\boldsymbol{x})=x_1^2$$
$$f_6(\boldsymbol{x})=x_2^2$$

此时，试验设计的具体回归模型就可写成

$$y=\beta_1+\beta_2x_1+\beta_3x_2+\beta_4x_1x_2+\beta_5x_1^2+\beta_6x_2^2+e \tag{4-20}$$

当试验计划进行 n 次，由式（4-19）表出的回归模型可得

$$y_i=\beta_1 f_1(\boldsymbol{x}_i)+\beta_2 f_2(\boldsymbol{x}_i)+\cdots+\beta_m f_m(\boldsymbol{x}_i)+e \quad (i=1,2,\cdots,n)$$

其试验设计矩阵 $\boldsymbol{X}$ 为

$$\boldsymbol{X}=\begin{bmatrix} f_1(\boldsymbol{x}_1) & f_2(\boldsymbol{x}_1) & \cdots & f_m(\boldsymbol{x}_1) \\ f_1(\boldsymbol{x}_2) & f_2(\boldsymbol{x}_2) & \cdots & f_m(\boldsymbol{x}_2) \\ \vdots & \vdots & & \vdots \\ f_1(\boldsymbol{x}_n) & f_2(\boldsymbol{x}_n) & \cdots & f_m(\boldsymbol{x}_n) \end{bmatrix} \tag{4-21}$$

此时，该试验设计的信息矩阵可定义为（$\boldsymbol{X}'\boldsymbol{X}$）。

这样，就可通过选择 $\boldsymbol{X}$ 矩阵而使由式（4-2）表出的回归模型可估，使由式（4-2）可估的设计称为可行设计。记所有可行设计的全体组成集合为 $\mathfrak{R}_n$。所谓最优试验设计就是指从 $\mathfrak{R}_n$ 中找出一个 $\boldsymbol{X}\in\mathfrak{R}_n$，使式（4-2）的最小二乘估计具有某种“优良性”。

“优良性”指什么？这涉及到不同的准则，下面从不同的角度介绍几种常用的优良性准则。

（1）A-最优准则　对任取 $\boldsymbol{X}\in\mathfrak{R}_n$ 有

$$\beta=(X^{t}X)^{-1}X^{t}y \tag{4-22}$$

而

$$\operatorname{cov}(\beta)=\operatorname{cov}[(X^{t}X)^{-1}X^{t}y]=\sigma^{2}(X^{t}X)^{-1} \tag{4-23}$$

在此，σ^2 为回归模型的估计误差的方差。若存在 $X_A\in\Re_n$，使

$$\operatorname{tr}[\operatorname{cov}(\beta)]=\operatorname{tr}[\sigma^{2}(X_{A}^{t}X_{A})^{-1}]=\min\sum_{i=1}^{m}1/\lambda_i \tag{4-24}$$

A-最优是协方差矩阵的特征值之和最小准则，或称平均方差最小准则。

(2) E-最优准则　若存在 $X_E\in\Re_n$，使得

$$\lambda_{\min}(X_{E}^{t}X_{E})=\max_{X\in\Re_n}\lambda_{\min}(X^{t}X) \tag{4-25}$$

成立，则称 X_E 为 E-最优的设计矩阵。其中 $\lambda_{\min}$为信息矩阵 ($X_E^tX_E$) 的最小特征值。故 E-最优是信息矩阵的最小特征值最大准则。

(3) D-最优准则　若存在 $X_D\in\Re_n$，使得

$$|X_{D}^{t}X_{D}|=\max_{X\in\Re_n}|X^{t}X| \tag{4-26}$$

成立，则称 X_D 为 D-最优的设计矩阵。其中 $|.|$ 为信息矩阵 (X^tX) 的行列式值。故 D-最优是信息矩阵的行列式值最大准则。

(4) G-最优准则　对任取 $X\in\Re_n$ 有

$$y=X\beta \quad 即 \quad y_i=x_i^{t}\beta=x_i^{t}(X^{t}X)^{-1}X^{t}y \tag{4-27}$$

所以有

$$\operatorname{var}(y_i)=\operatorname{var}(x_i^{t}(X^{t}X)^{-1}X^{t}y)=x_i^{t}(X^{t}X)^{-1}X^{t}\operatorname{var}(y)X(X^{t}X)^{-1}x_i=x_i^{t}(X^{t}X)^{-1}x_i\sigma^{2}$$

则

$$\operatorname{var}(y_i)=x_i^{t}(X^{t}X)^{-1}x_i\sigma^{2}$$

若存在 $X_g\in\Re_n$，使得

$$\mathrm{d}(X_g)=\min_{X\in\Re_n}\max_{x\in X}x^{t}(X^{t}X)^{-1}x$$

成立，则称 X_g 为 G-最优的设计矩阵。G-最优准则是使预报值在整个区域中的最大方差达到最小的设计，故也称 min max 设计。

上面都是假设试验次数 n 是固定的，通常称为离散型设计，记矩阵

$$M\equiv1/n(X^{t}X)\equiv(m_{ij})$$

为信息阵。以上介绍的各种最优准则实质上是讨论信息阵的某一函数 $f(M)$的最小值问题。设 $\Re$ 是 x 的取值域（试验区域），Kiefer 在 $\Re$ 引进一个测度 $\mu(x)$。定义信息阵 $M\equiv(m_{ij})$中的 m_{ij}为

$$m_{ij}=\int_{\Re}x_i x_j\,\mathrm{d}\mu(x) \tag{4-28}$$

给出了一个测度 $\mu(x)$就相当于给出一个设计，这样的设计称为连续型设计。Kiefer 证明了在连续型设计中 D-最优与 G-最优的等价性的著名定理。但在离散型设计中，有例子说明 D-最优可以不是 G-最优。在最优试验设计中，有实际应用意义的是离散型设计。

二、D-最优设计

对于式 (4-2) 表示的回归模型可以编制出各种试验设计，评价这些设计的好坏，可以从试验点的多少（如部分因子设计、均匀设计），也可以从计算简繁（如因子设计、正交设计

等）来评价。从上节的讨论可知，D-最优设计是从对模型的参数 $\boldsymbol{\beta}$ 的估计好坏来评价的。

对给定的模型，可以寻找一个试验设计 Ω，通过试验得到一组数据，然后根据最小二乘法来获得 $\boldsymbol{\beta}$ 的估计。对于不同的试验设计，可得到不同的最小二乘估计。为了评价这些参数估计的好坏，需要对这些参数的估计值进行综合考察，D-最优就是用它们的密集椭球体的体积的大小来评价的。

设所研究问题的回归模型为

$$y=\beta_1+\beta_2x+\beta_3x^2+e \tag{4-29}$$

在因子区域 $-1\leqslant x\leqslant 1$ 上比较下述两个试验设计

$$\Omega_1:\begin{cases}x_1=-1 \quad x_2=0 \quad x_3=1\\ n_1=5 \quad n_2=5 \quad n_3=5\end{cases}$$

$$\Omega_2:\begin{cases}x_1=-1 \quad x_2=0 \quad x_3=1\\ n_1=4 \quad n_2=7 \quad n_3=4\end{cases}$$

由此可得它们的试验设计矩阵 $\boldsymbol{X}$ 分别为

$$\boldsymbol{X}_1=\begin{bmatrix}1 & -1 & 1\\ 1 & -1 & 1\\ 1 & -1 & 1\\ 1 & -1 & 1\\ 1 & -1 & 1\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$$\boldsymbol{X}_2=\begin{bmatrix}1 & -1 & 1\\ 1 & -1 & 1\\ 1 & -1 & 1\\ 1 & -1 & 1\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 0 & 0\\ 1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

则它们的信息矩阵 $(X_1^tX_1)$、$(X_2^tX_2)$ 分别为

$$(X_1^tX_1)=\begin{bmatrix}15 & 0 & 10\\ 0 & 10 & 0\\ 10 & 0 & 10\end{bmatrix}$$

$$(X_2^tX_2)=\begin{bmatrix}15 & 0 & 8\\ 0 & 8 & 0\\ 8 & 0 & 8\end{bmatrix}$$

$$|(X_1^tX_1)|=15\times10\times10-10\times10\times10=5\times10\times10=500$$

$$|(X_2^tX_2)|=15\times8\times8-8\times8\times8=7\times8\times8=448$$

显然有 $|(X_1^tX_1)|\geqslant|(X_2^tX_2)|$，所以，在 D-优良性的意义下，设计 Ω_1 比 Ω_2 好。

三、等价定理及 D-最优设计的构造

在给定的因子区域 $\mathfrak{R}$ 上，直接用前述方法来研究和构造 D-最优设计，一般还是很困难的，因为在此因子区域 $\mathfrak{R}$ 上，可构造无数个设计，要用上述方法对它们进行一一比较，才能找到真正的 D-最优设计。为克服这一困难，人们又研究了与 D-最优设计有关的 G-最优设计的等价问题。在第二节中，曾给出了 G-最优准则，该准则的目标是使预报值在整个区域中的最大方差达到最小，即对于因子区域 $\mathfrak{R}$ 上的任意两个设计 Ω_1 和 Ω_2，若有

$$\max \mathrm{d}(X_1,\Omega_1)=\max \boldsymbol{f}^t(\boldsymbol{x})(X_1^tX_1)^{-1}\boldsymbol{f}(\boldsymbol{x})<\max \mathrm{d}(X_2,\Omega_2)=\max \boldsymbol{f}^t(\boldsymbol{x})(X_2^tX_2)^{-1}\boldsymbol{f}(\boldsymbol{x})$$

则在 G-最优意义下，设计 Ω_1 比 Ω_2 好。在此

$$\boldsymbol{f}(\boldsymbol{x})=\begin{pmatrix}f_1(\boldsymbol{x})\\ f_2(\boldsymbol{x})\\ \vdots\\ f_m(\boldsymbol{x})\end{pmatrix}$$

下面来看看由前节讨论的两个设计 Ω_1 和 Ω_2 的例子，对于由式（4-29）表出的回归模型，设计 Ω_1 的估计方差

$$\begin{aligned}
\mathrm{d}(X_1,\Omega_1)&=\boldsymbol{f}^t(\boldsymbol{x})(X_1^tX_1)^{-1}\boldsymbol{f}(\boldsymbol{x})\\
&=(1,x,x^2)\begin{bmatrix}15 & 0 & 10\\ 0 & 10 & 0\\ 10 & 0 & 10\end{bmatrix}^{-1}\begin{bmatrix}1\\ x\\ x^2\end{bmatrix}\\
&=(1,x,x^2)\begin{bmatrix}0.20 & 0 & -0.20\\ 0 & 0.10 & 0\\ -0.20 & 0 & 0.30\end{bmatrix}\begin{bmatrix}1\\ x\\ x^2\end{bmatrix}\\
&=(1,x,x^2)\begin{bmatrix}0.20-0.20x^2\\ 0.10x\\ -0.20+0.30x^2\end{bmatrix}\\
&=(0.20-0.20x^2)+0.10x^2+(-0.20+0.30x^2)x^2\\
&=0.3x^4-0.3x^2+0.2=0.1\times(3x^4-3x^2+2)
\end{aligned}$$

函数 $(3x^4-3x^2+2)$ 在因子区域 $-1\leqslant x\leqslant 1$ 上的最大值为 2，所以

$$\max \mathrm{d}(X_1,\Omega_1)=\max \boldsymbol{f}^t(\boldsymbol{x})(X_1^tX_1)^{-1}\boldsymbol{f}(\boldsymbol{x})=1/10\max_{-1\leqslant x\leqslant1}(3x^4-3x^2+2)=(1/10)\times2=0.2$$

而设计 Ω_2 的估计方差

$$
\begin{aligned}
d(X_2,\Omega_2) &= \boldsymbol{f}'(\boldsymbol{x})(X_1'X_1)^{-1}\boldsymbol{f}(\boldsymbol{x}) \\
&= (1,x,x^2)\begin{bmatrix}15 & 0 & 8\\ 0 & 8 & 0\\ 8 & 0 & 8\end{bmatrix}^{-1}\begin{bmatrix}1\\ x\\ x^2\end{bmatrix} \\
&= (1,x,x^2)\begin{bmatrix}0.1429 & 0 & -0.1429\\ 0 & 0.1250 & 0\\ -0.1429 & 0 & 0.2679\end{bmatrix}\begin{bmatrix}1\\ x\\ x^2\end{bmatrix} \\
&= (1,x,x^2)\begin{bmatrix}0.1429-0.1429x^2\\ 0.125x\\ -0.1429+0.2679x^2\end{bmatrix} \\
&= (0.1429-0.1429x^2)+0.125x^2+(-0.1429+0.2679x^2)x^2 \\
&= (0.2679x^4-0.1608x^2+0.1429)
\end{aligned}
$$

函数（$0.2679x^4-0.1608x^2+0.1429$）在因子区域$-1\leqslant x\leqslant 1$上的最大值为 0.1429、所以

$$\max d(X_2,\Omega_2)=\max \boldsymbol{f}'(\boldsymbol{x})(X_1'X_1)^{-1}\boldsymbol{f}(\boldsymbol{x})=\max_{-1\leqslant x\leqslant 1}(0.2679x^4-0.1608x^2+0.1429)=0.1429$$

因此，在 G-最优意义下，设计 Ω_1 仍比 Ω_2 好。

下面的等价定理告诉我们，D-最优设计与 G-最优设计是等价的，其证明可参见文献［20］。

等价定理　下面三个结论是相互等价的：

① 试验设计 Ω^* 是 D-最优的，即

$$|X(\Omega^*)'X(\Omega^*)|=\max_{\Omega}|X(\Omega^*)'X(\Omega^*)| \tag{4-30}$$

② 试验设计 Ω^* 是 G-最优的，即

$$\max_{f(x)}\boldsymbol{f}'(\boldsymbol{x})(X_1'X_1)^{-1}\boldsymbol{f}(\boldsymbol{x})=\min_{x}\max_{\Omega}\boldsymbol{f}'(\boldsymbol{x})(X_1'X_1)^{-1}\boldsymbol{f}(\boldsymbol{x}) \tag{4-31}$$

③ 试验设计 Ω^* 使得

$$\max_{x}\boldsymbol{f}'(\boldsymbol{x})(X_1'X_1)^{-1}\boldsymbol{f}(\boldsymbol{x})=m(\text{未知参数个数}) \tag{4-32}$$

③是由②得来的，它对于构造和检验 D-最优设计较为方便。下面给出一个例子，即应用等价定理来构造出一个设计（此例引自文献［20］），然后应用等价定理去验证其 D-最优性。

在 d 维立方体$-1\leqslant x_j\leqslant 1$，$j=1$，2，…，$d$ 上，对于 d 元二次回归模型

$$y=\beta_1f_1(\boldsymbol{x})+\beta_2f_2(\boldsymbol{x})+\cdots+\beta_mf_m(\boldsymbol{x})+e \tag{4-33}$$

式中

$$
\begin{aligned}
&m=(d+1)(d+2)/2\\
&\boldsymbol{f}_1(\boldsymbol{x})=1\\
&\boldsymbol{f}_{1+j}(\boldsymbol{x})=x_j^2 \qquad 1\leqslant j\leqslant d\\
&f_{p+1+j}(\boldsymbol{x})=x_j \qquad 1\leqslant j\leqslant d\\
&\boldsymbol{f}_k(\boldsymbol{x})=x_ix_j \qquad 2d+2\leqslant k\leqslant m \qquad 1\leqslant i<j\leqslant d
\end{aligned}
$$

设想构造这样的 D-最优设计 Ω^*，它的试验点选自 p 维立方体的 2^d 个顶点，每个点赋予测度 μ(参见前节的试验设计定义。实际上，试验点的测度一定，则其试验重复次数也就唯一确定了，试验设计也就唯一确定了)；$d\times 2^{d-1}$个棱的中点，每个点赋予测度 υ；$C_d^2\times 2^{d-1}$个二维而的中点，每个点赋予测度 τ。这样的设计 Ω 是由

$$N=2^{d-3}[8+4d+d(d-1)]$$

个试验点构成。例如 $d=3$ 时，这样的设计 Ω 是由 8 个顶点、12 条棱的中点和 6 个二维面的中心，共 26 个试验点构成（见图 4-10）。从图 4-10 可以看出，假如 d 维立方体中心就是因子空间的坐标原点，那么这个设计 Ω 的试验点的坐标就只能取 0，−1，1 三种情况。其中顶点的坐标都不为 0，棱的中点的坐标中只有一个为 0，二维面的中心的坐标有两个为 0。

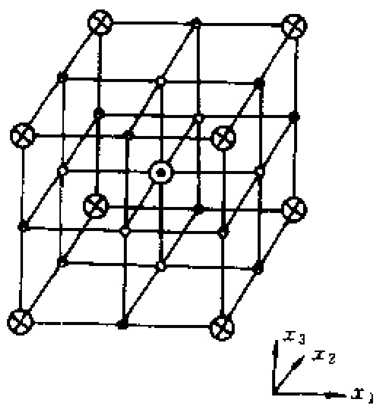


图 4-10　三因子设计的试验点分布

○—二维面的中心；◓—棱的中心；⊗—顶点；⊙—中心点

显然，要寻找这类 D-最优设计，就是要找出满足 D-最优性的测度 μ、υ、τ。可以分两步进行：第一步是在条件

$$\sum\mu+\sum\upsilon+\sum\tau=1$$

下，求出使 $[X(\Omega)'X(\Omega)]$ 达到极大值的 μ、υ、τ；进一步验证所求出的设计是否满足

$$\max_{f(x)} f'(x)(X_1'X_1)^{-1}f(x)=m$$

第一步：首先写出上述设计 Ω 的信息矩阵 $[X(\Omega)'X(\Omega)]$。

上述模型的信息矩阵可一般地表示为

$$[X(\Omega)'X(\Omega)]=\sum p_i f(x)f(x)'$$

$$=\begin{vmatrix} \sum p_i f_1^2(x) & \sum p_i f_1(x)f_2(x) & \cdots & \sum p_i f_1(x)f_m(x) \\ \sum p_i f_1(x)f_2(x) & \sum p_i f_2^2(x) & \cdots & \sum p_i f_2(x)f_m(x) \\ \vdots & \vdots & & \vdots \\ \sum p_i f_1(x)f_m(x) & \sum p_i f_1(x)f_m(x) & & \sum p_i f_m^2(x) \end{vmatrix}$$

式中

$$f(x)=\begin{vmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{vmatrix}$$

当 $d=3$ 时，上述设计 Ω 中的三类点及其二项式的取值列于表 4-25。从表 4-25 可见，对任意 i，j，$k=0$，1，2，3 都有

$$\sum_{t=1}^{26} p_t x_{tj} = 0 \quad (j=1,2,3)$$

$$\sum_{t=1}^{26} p_t x_{ti} x_{tj} = 0 \quad (i,j=1,2,3)$$

$$\sum_{t=1}^{26} p_t x_{ti}^2 x_{tj} = 0 \quad (j\neq 0)$$

$$\sum_{t=1}^{26} p_t x_{ti}^3 x_{tj} = 0$$

$$\sum_{t=1}^{26} p_t x_{ti} x_{tj} x_{tk} = 0$$

$$\sum_{t=1}^{26} p_t x_{ti}^2 x_{tj} x_{tk} = 0$$

而

$$\mu = \sum_{t=1}^{n} p_t x_{tj}^2 = \sum_{t=1}^{n} p_t x_{tj}^4 \neq 0$$

$$\upsilon=\sum_{t=1}^{n}p_t x_{ti}^2 x_{tj}^2 \neq 0$$

表 4-25 设计中三类点的坐标及其二次项的取值

实验点	测度 p_t	x_0	x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	x_1^2	x_2^2	x_3^2
第一类点：8个顶点	μ	1	1	1	1	1	1	1	1	1	1
		1	1	1	−1	1	−1	−1	1	1	1
		1	1	−1	1	−1	1	−1	1	1	1
		1	1	−1	−1	−1	−1	1	1	1	1
		1	−1	1	1	−1	−1	1	1	1	1
		1	−1	1	−1	−1	1	−1	1	1	1
		1	−1	−1	1	1	−1	−1	1	1	1
		1	−1	−1	−1	1	1	1	1	1	1
第二类点：12条棱的中点	υ	1	0	1	1	0	0	1	0	1	1
		1	0	1	−1	0	0	1	0	1	1
		1	0	−1	1	0	0	−1	0	1	1
		1	0	−1	−1	0	0	1	0	1	1
		1	1	0	1	0	1	0	1	0	1
		1	1	0	−1	0	−1	0	1	0	1
		1	−1	0	1	0	−1	0	1	0	1
		1	−1	0	−1	0	1	0	1	0	1
		1	1	1	0	1	0	0	1	1	0
		1	1	−1	0	−1	0	0	1	1	0
		1	−1	1	0	−1	0	0	1	1	0
		1	−1	−1	0	1	0	0	1	1	0
第三类点：6个二维面的中心	τ	1	0	0	1	0	0	0	0	0	1
		1	0	0	−1	0	0	0	0	0	1
		1	0	1	0	0	0	0	0	1	0
		1	0	−1	0	0	0	0	0	1	0
		1	1	0	0	0	0	0	1	0	0
		1	−1	0	0	0	0	0	1	0	0

如果在上述设计矩阵（表 4-24）中，把 x_j^2，$j=1$，2，3 移至 x_0 和 x_1 之间，那么它的信息矩阵为（空白处为零）

$$[X(\Omega)'X(\Omega)]=\begin{pmatrix}1&\mu&\mu&\mu&&&&&&\\ \mu&\mu&\upsilon&\upsilon&&&&&&\\ \mu&\upsilon&\mu&\upsilon&&&&&&\\ \mu&\upsilon&\upsilon&\mu&&&&&&\\ &&&&\mu&&&&&\\ &&&&&\mu&&&&\\ &&&&&&\mu&&&\\ &&&&&&&\upsilon&&\\ &&&&&&&&\upsilon&\\ &&&&&&&&&\upsilon\end{pmatrix} \tag{4-34}$$

信息矩阵$[X(\Omega)'X(\Omega)]$具有如此简单的形式，是因为在每一类点中，除了 x_0 与 x_j^2，x_i^2 与 x_j^2 间外，其他各列之间都存在着正交性。只要设 Ω 中的试验点及其测度满足上述规定，这一性质对任意正整数 d 都是成立的。因此，在一般情况下的信息矩阵有如下形式（空白处为

零)：

$$[\boldsymbol{X}(\Omega)'\boldsymbol{X}(\Omega)]=\begin{bmatrix}1 & \boldsymbol{u}' & & \\ \boldsymbol{u} & \boldsymbol{G} & & \\ & & \mu\boldsymbol{I}_d & \\ & & & \upsilon\boldsymbol{I}_{d(d-1)/2}\end{bmatrix} \tag{4-35}$$

式中，$\boldsymbol{I}_d$ 为 d 阶单位矩阵；$\boldsymbol{u}'=(\mu,\mu,\cdots,\mu)$为 d 维行向量；$\boldsymbol{G}$ 为 d 阶方阵，其对角线元素为 μ，其他元素为 υ。

根据试验点坐标的性质，可以算得

$$\begin{aligned}\mu &= \sum_{i=1}^{n} p_i x_{ij}^2 = \mu 2^d + \upsilon(d-1)2^{d-1} + \tau 2^{d-1}[C_d^2 - (d-1)] \\ &= 2^{d-3}\{8\mu+4\upsilon(d-1)+\tau[(d-1)(d-2)]\}\end{aligned}$$

$$\begin{aligned}\upsilon &= \sum_{i=1}^{n} p_i x_{ih}^2 x_{ij}^2 \neq 0 \\ &= \mu 2^d+\upsilon(d-2)2^{d-1}+\tau 2^{d-2}[C_d^2-(d-1)-(d-2)] \\ &= 2^{d-3}\{8\mu+4\upsilon(d-2)+\tau[(d-2)(d-3)]\}\end{aligned}$$

通过计算得到矩阵$[\boldsymbol{X}(\Omega)'\boldsymbol{X}(\Omega)]$的行列式

$$|[\boldsymbol{X}(\Omega)'\boldsymbol{X}(\Omega)]|=\mu^d\upsilon^{d(d-1)/2}(\mu-\upsilon)^{d-1}[\mu+(d-1)\upsilon-d\upsilon^2] \tag{4-36}$$

对 μ 和 υ 求$|[\boldsymbol{X}(\Omega)'\boldsymbol{X}(\Omega)]|$的极大值，也就是在条件 $\mu>0,\upsilon>0$ 下解方程组

$$\partial|[\boldsymbol{X}(\Omega)'\boldsymbol{X}(\Omega)]|/\partial\mu=(2d+1)\mu^3-[(d+2)\upsilon+2]\mu^2-2(p-2)\mu\upsilon+(p-1)\upsilon^2=0$$

$$\partial|[\boldsymbol{X}(\Omega)'\boldsymbol{X}(\Omega)]|/\partial\upsilon=(d+1)\upsilon^2-[(d-2)\mu+(p+2)\mu^2]\upsilon+(p\mu-1)\mu^2=0$$

解得

$$\mu=\{(d+3)/[8(d+1)(d+2)^2]\}[(2d^2+3d+7)+(d-1)(4d^2+12d+17)^{1/2}]$$

$$\upsilon=\{(d+3)/[8(d+1)(d+2)^2]\}[(4d^3+8d^2+11d-5)+(2d^2+d+3)(4d^2+12d+17)^{1/2}]$$

另一方面，在式(4-37)条件下

$$\sum\mu+\sum\upsilon+\sum\tau=2^{d-3}[8\mu+4d\upsilon+d(d-1)\tau]=1 \tag{4-37}$$

解 μ 和 υ 的方程组得

$$\mu=2^{-d-1}[(d-1)(d-2)-2d(d-2)\mu+d(d-1)\upsilon] \tag{4-38}$$

$$\upsilon=2^{-d+1}[(2d-3)\mu-(d-1)\upsilon-(d-2)] \tag{4-39}$$

$$\tau=2^{2-d}[1+\upsilon-2\mu] \tag{4-40}$$

再把 μ 和 υ 的值代入可得

$$\begin{aligned}\mu=&[2^{d+4}(d+2)^3(d+1)]^{-1}\{(4d^6+12d^5-25d^4-107d^3+85d^2+479d+128)- \\ &(2d^2-d-19)d(d-1)(d+3)(4d^2+12d+17)^{1/2}\}\end{aligned} \tag{4-41a}$$

$$\begin{aligned}\upsilon=&[2^{d+2}(d+2)^3(d+1)]^{-1}\{-(4d^5+16d^4-11d^3-143d^2-149d+139)+ \\ &(2d^2+d-15)(d-1)(d+3)(4d^2+12d+17)^{1/2}\}\end{aligned} \tag{4-41b}$$

$$\begin{aligned}\tau=&[2^{d+1}(d+2)^3(d+1)]^{-1}\{(4d^4+24d^3+43d^2-24d-119)- \\ &(2d^2+3d-11)(d+3)(4d^2+12d+17)^{1/2}\}\end{aligned} \tag{4-41c}$$

第二步：先求出矩阵$[\boldsymbol{X}(\Omega)'\boldsymbol{X}(\Omega)]$的逆矩阵

$$[\boldsymbol{X}(\Omega)'\boldsymbol{X}(\Omega)]^{-1}=\begin{bmatrix}a & \boldsymbol{b}' & & \\ \boldsymbol{b} & \boldsymbol{E} & & \\ & & \mu^{-1}\boldsymbol{I}_d & \\ & & & \upsilon^{-1}\boldsymbol{I}_{d(d-1)/2}\end{bmatrix} \tag{4-42}$$

若令

$$h=[\mu+(d-1)\upsilon-d\mu^2]^{-1}$$
$$s=[\mu+(d-2)\upsilon-(d-1)\mu^2]/(\mu-\upsilon)$$

则

$$a=[\mu+(d-1)]h$$
$$\boldsymbol{b}'=(-\mu h,-\mu h,\cdots,-\mu h)$$

$\boldsymbol{E}$ 为 $d\times d$ 阶方阵

$$\boldsymbol{E}=\begin{pmatrix} sh & [(\mu^2-\upsilon)/(\mu-\upsilon)]h & \cdots & [(\mu^2-\upsilon)/(\mu-\upsilon)]h \\ [(\mu^2-\upsilon)/(\mu-\upsilon)]h & sh & \cdots & [(\mu^2-\upsilon)/(\mu-\upsilon)]h \\ \vdots & \vdots & & \vdots \\ [(\mu^2-\upsilon)/(\mu-\upsilon)]h & [(\mu^2-\upsilon)/(\mu-\upsilon)]h & & sh \end{pmatrix} \tag{4-43}$$

将 μ 和 υ 的值代入 $d(X_1,\Omega_1)=\boldsymbol{f}'(\boldsymbol{x})(X_1'X_1)^{-1}\boldsymbol{f}(\boldsymbol{x})$ 中的 $(X_1'X_1)^{-1}$，得到

$$d(X_1,\Omega_1)=\boldsymbol{f}'(\boldsymbol{x})(X_1'X_1)^{-1}\boldsymbol{f}(\boldsymbol{x})=(d+1)(d+2)/2-c\sum_{j=1}^{d}(x_j^2-x_j^4)$$

这里 c 为一正常数，不难看到在因子区域 $-1\leqslant x_j\leqslant 1, j=1,2,\cdots,d$ 上

$$\max_{f(x)} d(X_1,\Omega_1)=\boldsymbol{f}'(\boldsymbol{x})(X_1'X_1)^{-1}\boldsymbol{f}(\boldsymbol{x})=(d+1)(d+2)/2=m$$

根据等价定理知道，上面求得的 μ、υ、τ 的值是满足 D-最优设计条件的，也就是说，μ、υ、τ 是最优测度，它们构成的设计是 D-最优设计。

在 $d\leqslant5$ 的情况下，根据式（4-41）算出的 μ、υ、τ 的值列于表 4-26 中。

在 $d\geqslant6$ 时，上面这种 D-最优设计的构造方法是不适合的，因为此时 $\upsilon<0$，而在实际上是不可能的。

表 4-26　$d\leqslant5$ 时 μ、υ、τ 的值

d	μ	υ	τ	d	μ	υ	τ
1	0.333	0.333	0.000	4	0.03705	0.00384	0.01185
2	0.1458	0.08015	0.0962	5	0.01926	0.003125	0.004475
3	0.071975	0.01895	0.03280				

四、构造 D-最优设计的数值方法

从前述立方体设计的例子可以看出，D-最优设计是先通过直觉的方法给出试验点，继求出在这些试验点上满足 D-最优性的测度（即出现的概率），最后用等价定理进行验证，这些构造 D-最优设计的方法，关键在给定的因子区域 $\mathfrak{R}$ 中合理地选择试验点和规定测度，使设计的信息矩阵达到最优。然而，这样的方法只是在一些最简单情况下才是可行的，一般情况下用这种方法直接构造 D-最优设计是很困难的。所以人们又转向用数值方法来构造 D-最优设计。所谓数值方法，就是在给定区域 $\mathfrak{R}$ 和已知函数列 $f(x)$ 的情况下，用最优化方法寻找使信息矩阵行列式

$$\boldsymbol{A}(\Omega)=|[\boldsymbol{X}(\Omega)'\boldsymbol{X}(\Omega)]|=\left|\sum_{i=1}^{n}p_i\boldsymbol{f}(x_i)\boldsymbol{f}'(x_i)\right|=L(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n,p_1,p_2,\cdots,p_n) \tag{4-44}$$

在条件

$$p_i\geqslant0\qquad\sum_{i=1}^{n}p_i=1$$

下达到最大的 x_i，p_i（$i=1, 2, \cdots, n$）的值。

在此，所谓函数列 $\boldsymbol{f}(\boldsymbol{x})$有如下定义，如有下述的回归模型：

$$y=\beta_1+\beta_2x_1+\beta_3x_2+\beta_4x_1x_2+\beta_5x_1^2+\beta_6x_2^2+e \tag{4-45}$$

则 $\boldsymbol{f}(\boldsymbol{x})$ 具有如下形式

$$\boldsymbol{f}(\boldsymbol{x})=\begin{pmatrix}1\\x_1\\x_2\\x_1x_2\\x_1^2\\x_2^2\end{pmatrix} \tag{4-46}$$

下面简单介绍一种寻找 D-最优设计的数值方法。为了方便，仍用模型（4-2）式来进行讨论，在介绍 D-最优设计的数值方法之前，先介绍两个基本概念。

（一）线性组合设计

设

$$\Omega_1\begin{cases}\boldsymbol{x}_{11},\boldsymbol{x}_{12},\cdots,\boldsymbol{x}_{1n}\\p_1,p_2,\cdots,p_n\end{cases},\quad \Omega_2\begin{cases}\boldsymbol{x}_{21},\boldsymbol{x}_{22},\cdots,\boldsymbol{x}_{2m}\\q_1,q_2,\cdots,q_m\end{cases}$$

是两个规范设计，则按试验设计的定义，有

$$p_i\geqslant 0\qquad \sum_{i=1}^{n}p_i=1;\quad q_j\geqslant 0\qquad \sum_{j=1}^{m}q_j=1$$

如果 α 是[0,1]区间上的数，则

$$\sum_{i=1}^{n}(1-\alpha)p_i+\sum_{j=1}^{m}\alpha q_j=(1-\alpha)\sum_{i=1}^{n}p_i+\alpha\sum_{j=1}^{m}q_j=(1-\alpha)+\alpha=1$$

因此，由设计 Ω_1 和 Ω_2 得到的设计

$$\Omega\begin{cases}\boldsymbol{x}_{11},\boldsymbol{x}_{12},\cdots,\boldsymbol{x}_{1n} & \boldsymbol{x}_{21},\boldsymbol{x}_{22},\cdots,\boldsymbol{x}_{2m}\\(1-\alpha)p_1,(1-\alpha)p_2,\cdots,(1-\alpha)p_n & \alpha q_1,\alpha q_2,\cdots,\alpha q_m\end{cases}$$

仍是一规范设计，并称 Ω 为 Ω_1、Ω_2 的线性组合设计，记之为

$$\Omega=(1-\alpha)\Omega_1+\alpha\Omega_2 \tag{4-47}$$

值得指出的是，线性组合设计 Ω 的信息矩阵 $\boldsymbol{A}(\Omega)$与 Ω_1、Ω_2 的信息矩阵 $\boldsymbol{A}(\Omega_1)$和 $\boldsymbol{A}(\Omega_2)$的关系为

$$\boldsymbol{A}(\Omega)=(1-\alpha)\boldsymbol{A}(\Omega_1)+\alpha\boldsymbol{A}(\Omega_2) \tag{4-48}$$

（二）一点设计

只含一个试验点 $\boldsymbol{x}_0$ 的设计成为一点设计，记之为 $\Omega(\boldsymbol{x}_0)$，显然，试验点 $\boldsymbol{x}_0$ 的测度为$p_0=1$。设 $\Omega_a(\boldsymbol{x})$为任意设计

$$\Omega_a\begin{cases}\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\\p_1,p_2,\cdots,p_n\end{cases}$$

并设 $\Omega_b(\boldsymbol{x})$是 $\Omega_a(\boldsymbol{x})$与 $\Omega(\boldsymbol{x}_0)$的线性组合设计，即 $\Omega_b(\boldsymbol{x})$为

$$\Omega_b\begin{cases}\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n & \boldsymbol{x}_0\\(1-\alpha)p_1,(1-\alpha)p_2,\cdots,(1-\alpha)p_n & \alpha\end{cases}$$

则由(4-48)式，可知

$$A(\Omega_b)=(1-\alpha)A(\Omega_a)+\alpha A(\Omega(\boldsymbol{x}_0))=(1-\alpha)A(\Omega_a)+\alpha \boldsymbol{f}(\boldsymbol{x}_0)\boldsymbol{f}(\boldsymbol{x}_0)^{\tau} \tag{4-49}$$

根据行列式的性质(参见第十一章),可得

$$\begin{aligned}|A(\Omega_b)|&=(1-\alpha)^m|A(\Omega_a)+[\alpha/(1-\alpha)]\boldsymbol{f}(\boldsymbol{x}_0)\boldsymbol{f}(\boldsymbol{x}_0)^{\tau}|\\&=(1-\alpha)^m\{1+[\alpha/(1-\alpha)]d(\boldsymbol{x}_0,\Omega_0)\}\ A(\Omega_a)|\end{aligned}$$

式中,$d(\boldsymbol{x}_0,\Omega_0)$为设计 $\Omega(\boldsymbol{x}_0)$的方差函数在点 $\boldsymbol{x}_0$ 的值。

(三) 一个构造 D-最优设计的数值方法

从前面的讨论可知,在同一回归模型下,如果对于两个不同的设计 Ω_b 和 Ω_a,如有 $|A(\Omega_b)|>|A(\Omega_a)|$,则在 D-最优意义下,设计 Ω_b 比 Ω_a 好。那么,对于任一设计 Ω_a 与一点设计 $\Omega(\boldsymbol{x}_0)$的组合设计 Ω_b,怎样才能使 Ω_b 比 Ω_a 好呢?事实上,只要点 $\boldsymbol{x}_0$ 是设计 $\Omega(\boldsymbol{x}_0)$的方差函数的最大值点,即 $\boldsymbol{x}_0$ 满足

$$d(\boldsymbol{x}_0,\Omega_0)=\max_{x} d(\boldsymbol{x},\Omega_0)$$

并取组合系数 α 为

$$\alpha=[d(\boldsymbol{x}_0,\Omega_0)-m]/\{m[d(\boldsymbol{x}_0,\Omega_0)-1]\} \tag{4-50}$$

则必有

$$|A(\Omega_b)|>|A(\Omega_a)|$$

即设计 Ω_b 比 Ω_a 好,并且在所有的线性组合设计$(1-\alpha)\Omega_a+\alpha\Omega(\boldsymbol{x}_0)$中,$|A(\Omega_b)|$是信息矩阵行列式值最大的。

按上述方法,由初始设计 Ω_a 可得到一新设计 Ω_b,如果设计 Ω_b 还不是 D-最优的,那么,在 Ω_b 的基础上,再找到一点 $\boldsymbol{x}_1$,使得

$$d(\boldsymbol{x}_1,\Omega_1)=\max_{x} d(\boldsymbol{x},\Omega_1) \tag{4-51}$$

并取组合系数 α_1 为

$$\alpha_1=[d(\boldsymbol{x}_1,\Omega_1)-m]/\{m[d(\boldsymbol{x}_1,\Omega_1)-1]\} \tag{4-52}$$

再构造一个设计 Ω_b 与一点设计 Ω_1 的线性组合

$$\Omega_c=(1-\alpha)\Omega_b+\alpha_1\Omega_1 \tag{4-53}$$

则有 $\qquad |A(\Omega_c)|>|A(\Omega_b)|$

即 Ω_c 比 Ω_b 好,如果设计 Ω_c 还不是 D-最优的,又可按上述方法找到一个 Ω_d,不断重复上述过程,结果有两种可能,一是在第 k 步得到的设计 Ω_k 满足 $\max d(\boldsymbol{x},\Omega_k)=m$,即 Ω_k 是 D-最优的,另一种可能是这一过程无限重复进行,得到一系列设计

$$\Omega_a,\Omega_b,\Omega_c,\cdots,\Omega_k,\cdots$$

并且有

$$|A(\Omega_a)|<|A(\Omega_b)|<|A(\Omega_c)|<\cdots<|A(\Omega_k)|<\cdots \tag{4-54}$$

可以证明上述系列是收敛的,即

$$\lim_{k\to\infty}|A(\Omega_k)|=|A(\Omega^*)| \tag{4-55}$$

式中,Ω^*为 D-最优设计。

综上所述,用数值方法构造 D-最优设计的步骤如下:

① 给出一个初始设计 Ω_0

$$\Omega_0\begin{cases}\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\\ p_1,p_2,\cdots,p_n\end{cases}$$

其中

$$p_i \geqslant 0, \sum_{i=1}^{n} p_i = 1, \text{且 } |A(\Omega_0)| > 0$$

置 $k=0$

② 计算出信息矩阵 $A(\Omega_0)$及$|A(\Omega_0)|$

③ 在因子区域 $\mathfrak{R}$ 上,求出方差函数 $d(x,\Omega_k)$的最大值

$$d=d(x_k,\Omega_k)=\max_x d(x,\Omega_k) \tag{4-56}$$

式中

$$d(x,\Omega_k)=f^t(x)[X(\Omega_k)^tX(\Omega_k)]^{-1}f(x)$$

④ 如果 $d-m<\Delta$(Δ 为一预置精度,一般取 $\Delta=0.05$),则停止迭代,Ω_k 就是要寻找的近似 D-最优设计,否则计算

$$\alpha_k=(d-m)/[m(d-1)]$$

⑤ 作设计 Ω_k 与一点设计 $\Omega(x_k)$的线性组合

$$\Omega_{k+1}=(1-\alpha_k)\Omega_k+\alpha_k\Omega(x_k)$$

得新设计 Ω_{k+1},并计算出

$$A(\Omega_{k+1})=(1-\alpha_k)A(\Omega_k)+\alpha_kA[\Omega(x_k)]$$

$$|A(\Omega_{k+1})|=(1-\alpha_k)^m\{1+[\alpha_k/(1-\alpha_k)]d\}|A(\Omega_k)|$$

置 $k=k+1$

⑥ 重复进行③至⑤步,直至 $d-m<\Delta$。

用上述算法,可算得六维因子空间的二次 D-最优设计。

五、饱和 D-最优设计

一般说来,比较理想的试验设计除满足某种优良性外,还要求它所包含的实验点数越少越好。在试验设计中,如实验点数等于所要确定的未知参数的个数,则称该设计为饱和设计。值得提出的是,由于试验点数与未知参数的个数相等,故不存在剩余自由度,除非进行重复试验,否则无法对回归方程进行显著性检验。

(一) 一次饱和 D-最优设计

对于 s 维立方体 $-1\leqslant x_j\leqslant 1$,($j=1$, 2, …, s) 上的一次回归模型

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3+\cdots+\beta_sx_s$$

存在这样一个定理:

在 s 维立方体上选取 ($s+1$) 个点以组成一次饱和 D-最优设计时,只要考虑选取各个坐标都为 -1 或 1 的那些点。

在这个定理的基础上,用计算机可很容易找到 s 维立方体的一次饱和 D-最优设计:

当 $s=2$ 时,正方形区域的任何 3 个不同顶点都可组成 D-最优设计

当 $s=3$ 时,立方体区域上由 2^{3-1}个部分顶点所构成的试验设计,是 D-最优设计

当 $s=4$ 时,D-最优设计是

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix}$$

当 $s=5$ 时，D-最优设计是

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 \end{pmatrix}$$

或

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ -1 & -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & 1 \\ -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 \end{pmatrix}$$

当 $s=6$ 时，D-最优设计是

$$\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

当 $s=7$ 时，立方体区域上有 2^{7-4}个部分顶点所构成的试验设计，是 D-最优设计。

由于在饱和设计中，结构矩阵 $\boldsymbol{X}$ 是一个方阵，所以信息矩阵

$$\boldsymbol{A}(\Omega)=|\boldsymbol{X}'\boldsymbol{X}|=|\boldsymbol{X}|^2 \tag{4-57}$$

$\boldsymbol{A}(\Omega)$最大，也就是$|\boldsymbol{X}|$最大。一般说来，当 $(s+1)$ 是 2 的整数次幂时，s 个因子的一次饱和 D-最优设计，就可根据本章第一节所讨论的部分因子设计方案来进行，

（二）二次饱和 D-最优设计

对于二次回归模型

$$y=\beta_0+\sum_{j=1}^{s}\beta_j x_j+\sum_{i\leqslant j}^{s}\beta_{ij}x_i x_j \tag{4-58}$$

待估参数的个数为$(1+s+s+C_s^2)=[1+2s+s(s-1)/2]=(2+3s+s^2)=(s+1)(s+2)/2$ 个，那么存在不存在含试验点个数就为 $m=(s+1)(s+2)/2$ 的饱和 D-最优设计呢？有人已经证明，对 $s\geqslant7$ 不存在饱和 D-最优设计。

当 $s=2,3$ 时，饱和 D-最优设计列于表 4-27，表 4-27 中还列出了 $s=2$ 时的 7 点和 8 点的非饱和 D-最优设计。

关于 $s\geqslant4$ 的饱和 D-最优设计，至今尚未解决。对于 $s=4$，有人找到了一个较好的 15 点设

计，示于表 4-28。在表 4-28 中

$$f=g=-1 \qquad \alpha=\beta=-0.25$$

$$\gamma=-0.60 \qquad \delta=-h=0.05$$

这时
$$A(\Omega)=|X(\Omega)'X(\Omega)|=0.344547\times10^{-5}$$

表 4-27　$s=2,3$ 时的饱和 D-最优设计

试验号	$s=2$						$s=3$		
	6 点计划		7 点计划		8 点计划		10 点计划		
	x_1	x_2	x_1	x_2	x_1	x_2	x_1	x_2	x_3
1	−1	−1	−1	−1	−1	−1	−1	−1	−1
2	1	−1	1	−1	1	−1	1	−1	−1
3	−1	1	−1	1	−1	1	−1	1	−1
4	$-\delta$	δ	1	1	1	1	−1	−1	1
5	1	3δ	−0.092	0.092	1	0	−1	α	α
6	3δ	1	1	−0.067	0.032	1	α	−1	α
7			0.067	−1	0.032	−1	α	α	−1
8	$\delta=(4-\sqrt{3})/3$				−0.216	0	β	1	1
9	$=0.1315$						1	β	1
10							1	1	β
							$\alpha=0.1925$		
							$\beta=-0.2912$		

表 4-28　$s=4$ 的 15 点设计

试验号	x_1	x_2	x_3	x_4	试验号	x_1	x_2	x_3	x_4
4	−1	α	α	α	1	f	1	h	−1
6	β	δ	1	−1	13	f	1	−1	h
11	β	−1	δ	1	3	−1	−1	−1	−1
10	β	1	−1	δ	5	1	1	−1	−1
14	1	γ	1	1	2	1	−1	1	−1
15	1	1	γ	1	9	1	−1	−1	1
8	1	1	1	γ	12	g	1	1	1
7	f	h	−1	1					

根据 $s=2,3$ 时的饱和 D-最优设计的试验点结构，得到一般的二次饱和设计的方案如表 4-29。对于表 4-29 给出的设计，其信息矩阵行列式

$$A(\Omega)=|X(\Omega)'X(\Omega)|=|X(\Omega)|^2 \tag{4-59}$$

同时
$$|X(\Omega)|=2^s(\lambda+1)^{s(s-1)}[(\mu^2-1)-2(\lambda-1)(\mu-1)(s-2)/(\lambda+1)]^{(s-1)}$$
$$[(\mu^2-1)-4(\lambda-1)(s-1)(s+\mu-2)/(\lambda+1)] \tag{4-60}$$

求式(4-60)的极大值，只要解方程组

$$\partial|X(\Omega)|/\partial\lambda=0$$

$$\partial|X(\Omega)|/\partial\mu=0$$

求得使 $|X(\Omega)|$ 达到极大的 λ 和 μ。上面的方程组消去 λ 后，得到一个 μ 的 9 次多项式，这 9 次多项式的 9 个根，除了 3 个整根

$$\mu=-1,\mu=2-s,\mu=3-2s$$

外，其他 6 个根中只有一个在 s 维立方体，$-1\leqslant x_j\leqslant1, j=1,2,\cdots,s$ 内，对 $s\leqslant15$，算出的 λ、μ 的值列在表 4-30 中。从此表可以看出，当 $s=2,3$ 时，λ、μ 的值就是二次饱和 D-最优设计的参数值。

表 4-29　s 个因子的二次饱和设计

试 验 类 别	x_1	x_2	x_3	…	x_4
一个点	−1	−1	−1		−1
s 个点(1 个坐标是 1，$s-1$ 个坐标是 −1)	1	−1	−1		−1
	−1	1	−1		1
	…	…	…		…
	−1	−1	−1		1
s 个点(1 个坐标是 μ，$s-1$ 个坐标是 1)	μ	−1	−1		−1
	−1	μ	1		−1
	…	…	…		…
	−1	−1	1		μ
$\frac{s(s-1)}{2}$个点(2 个坐标是 λ，$p-2$ 个坐标是 −1)	λ	λ	−1		−1
	λ	1	λ		−1
	…	…	…		…
	−1	−1	−1		λ

表 4-30　$s\leqslant 15$ 的 λ、μ 的值

维数 s	μ	λ	维数 s	μ	λ
2	0.3944	−0.1315	9	−0.9602	0.7544
3	−0.2912	0.1925	10	−0.9693	0.7803
4	−0.6502	0.4114	11	−0.9757	0.8022
5	−0.8103	0.5355	12	0.9802	0.8198
6	−0.8354	0.6103	13	−0.9836	0.8346
7	0.9242	0.6772	14	−0.9862	0.8471
8	−0.9464	0.7208	15	−0.9882	0.8579

六、几种常用的 D-最优混合设计表

表 4-31～表 4-35 是 3 个 3 因素和 2 个 4 因素的 D-最优设计表，其中两个 3 因素的分别为 10 次试验和 11 次试验用的，而两个 4 因素的都是为 16 次试验用的。

表 4-31　3 因素 10 次 D-最优设计表

x_0	x_1	x_2	x_3	x_0	x_1	x_2	x_3
1	0	0	1.291	1	1	1	0.639
1	0	0	−0.136	1	1.174	0	−0.927
1	−1	−1	0.639	1	−1.174	0	0.927
1	1	−1	0.639	1	0	1.174	−0.927
1	−1	1	0.639	1	0	−1.174	−0.927

表 4-32　3 因素 11 次 D-最优设计表（一）

x_0	x_1	x_2	x_3	x_0	x_1	x_2	x_3
1	0	0	2	1	2	2	−1
1	0	0	−2	1	−2	−2	−1
1	−1.414	−1.414	1	1	0	0	−1
1	1.414	−1.414	1	1	0	0	−1
1	−1.414	1.414	1	1	0	0	0
1	1.414	1.414	1				

表 4-33　3 因素 11 次 *D*-最优设计表（二）

x_0	x_1	x_2	x_3	x_0	x_1	x_2	x_3
1	0	0	2.450	1	0.751	2.106	−1
1	0	0	−2.450	1	2.106	−0.751	−1
1	−0.751	2.106	1	1	−0.751	−2.106	−1
1	2.106	0.751	1	1	−2.106	0.751	−1
1	0.751	−2.106	1	1	0	0	0
1	−2.106	−0.751	1				

表 4-34　4 因素 16 次 *D*-最优设计表（一）

x_0	x_1	x_2	x_3	x_4	x_0	x_1	x_2	x_3	x_4
1	0	0	0	1.7884	1	−1	1	1	0.644
1	0	0	0	−1.784	1	1	1	1	0.644
1	−1	−1	−1	0.644	1	1.685	0	0	−0.908
1	1	−1	−1	0.644	1	−1.685	0	0	−0.908
1	−1	1	−1	0.644	1	0	1.685	0	−0.908
1	1	1	−1	0.644	1	0	−1.685	0	−0.908
1	−1	−1	1	0.644	1	0	0	1.685	−0.908
1	1	−1	1	0.644	1	0	0	−1.685	−0.908

表 4-35　4 因素 16 次 *D*-最优设计表（二）

x_0	x_1	x_2	x_3	x_4	x_0	x_1	x_2	x_3	x_4
1	0	0	0	1.7884	1	−1	1	1	0.644
1	0	0	0	−1.784	1	1	1	1	0.644
1	−1	−1	−1	0.644	1	1.685	0	0	−0.908
1	1	−1	−1	0.644	1	−1.685	0	0	−0.908
1	−1	1	−1	0.644	1	0	1.685	0	−0.908
1	1	1	−1	0.644	1	0	−1.685	0	−0.908
1	−1	−1	1	0.644	1	0	0	1.685	−0.908
1	1	−1	1	0.644	1	0	0	−1.685	−0.908

第七节　单纯形试验设计法

单纯形试验设计法是一种序贯试验设计方法，既可用来进行试验设计又可直接用于试验寻优，它由 Ernst[21]，在化学计量学中成为引人注目的试验优化方法，Deming 等有详细综述[22]。原始的简单单纯形法经过许多改进，如 Nelder 等改进的单纯形法[23]等在化学试验中应用较广。

在 s 维空间 R^s 中，单纯形是指具有 $s+1$ 个顶点的多面体，若各个棱长彼此相等，则称为正规单纯形。在二维空间中，$s+1=3$，即三角形是单纯形，等边三角形是正规单纯形。单纯形试验设计法是以单纯形顶点的坐标表述为试验各因素的水平取值，在二维空间中的三角形顶点坐标（x_1，x_2）表述的是两个因素的相应水平。如有 s 个因素，则用 s 维空间的单纯形顶点表述这 s 个因素的不同水平，按照起始单纯形的 $s+1$ 个顶点的坐标来安排 $s+1$ 个试验，然后通过比较这些试验结果，淘汰其中指标值最差的试验点，在可能改进试验指标的方向新增一个试验点，再作一次试验，继续单纯形搜索。

首先讨论起始单纯形的选取。设 $\{u_1, u_2, \cdots, u_s, u_{s+1}\}$ 是 R^s 中某一单纯形 $s+1$ 个顶点的位置矢量。拟用 u_i 表示试验中 s 个因素的水平，令 s 个因素各取某一起始水平的点为 x_0，这个起始点可根据化学知识与经验选取。对于每一个因素，根据化学家的经验选定一个步长。步

长代表的是考虑每一因素的影响时，从起始水平移动的幅度。例如选定起始 pH 为 7.0，如选择步长为 0.5，即是从 7.0 出发按 0.5 间距改变 pH 作试验。经过变换，可将原为不同数值与单位的各因素的步长化为相同的数值 α。可以证明，对给定的某一个顶点 x_0 和正数 α，按公式（4-61）取定的单纯形是以 x_0 为顶点且棱长为 α 的正规单纯形

$$v_1=x_0 \tag{4-61a}$$

$$v_i=x_0+z(i) \quad (i=2,3,\cdots,s+1) \tag{4-61b}$$

式中，$z(i)$为 s 维矢量

$$z(i)=[q,\cdots,\overset{i-1}{q},p,q,\cdots,q]^t \tag{4-62}$$

式(4-62)中

$$p=(\sqrt{s+1}+s-1)[\alpha/(s\sqrt{2})] \tag{4-63a}$$

$$q=(\sqrt{s+1}-1)[\alpha/(s\sqrt{2})] \tag{4-63b}$$

现计算 v_i 与 v_1 的距离

$$\| v_i-v_1 \|^2=\| x_0+z(i)-x_0 \|^2=\| z(i) \|^2=(s-1)q^2+p^2 \tag{4-64}$$

式(4-63a)、(4-63b)代入式(4-64)后得

$$\begin{aligned}\| v_i-v_1 \|^2 &=(s-1)(\sqrt{s+1}-1)^2[\alpha/(s\sqrt{2})]^2+(\sqrt{s+1}+s-1)^2[\alpha/(s\sqrt{2})]^2\\ &=[\alpha^2/(s^2 2)]\{(s-1)(s+1-2\sqrt{s+1}+1)+\\ &\quad [s+1+2\sqrt{s+1}(s-1)+(s-1)^2]\}\\ &=[\alpha^2/(s^2 2)]\{(s-1)s+2(s-1)+s+1+(s-1)^2\}\\ &=[\alpha^2/(s^2 2)](s^2-s+2s-2+s+1+s^2-2s+1)\\ &=[\alpha^2/(s^2 2)](2s^2)\\ &=\alpha^2\end{aligned}$$

可见这个有 $s+1$ 个顶点的单纯形各棱长均为 α。令 $v_1=[v_1,v_2,\cdots,v_s]^t$，则各个顶点的坐标可分别为

$$\begin{aligned}v_2&=[v_1+p,v_2+q,\cdots,v_s+q]^t\\ v_3&=[v_1+q,v_2+p,\cdots,v_s+q]^t\\ &\vdots\\ v_{s+1}&=[v_1+q,v_2+q,\cdots,v_s+p]^t\end{aligned}$$

表 4-36　计算起始单纯形的 p、q 值

s	p	q
2	0.966α	0.259α
3	0.943α	0.236α
4	0.926α	0.219α
5	0.911α	0.204α
6	0.901α	0.194α

可见只要根据经验确定了一个可行的起始点 v_1 和步长 α，就能开始试验。关于 p、q 值的选取可容地根据式（4-63）计算，表 4-36 给出了几个例子。

设试验的目的是搜索指标取值最大的最佳点。比较各 v_i 的指标取值，设 v_w 是各点中的最差点，则进行反射，去掉最坏点 v_w，用其对称点作新试验点。先求去掉 v_w 后余下各点的形心 v_g

$$v_g=(1/s)\sum_{\substack{i=1\\ i\neq w}}^{s+1} v_i \tag{4-65}$$

反射是按如公式(4-66)求发射点 v_r

$$v_f = v_g + \beta(v_g - v_w) \tag{4-66}$$

当 $s=2$ 时，在二维平面上绘出有关图形，见图 4-11。取 $\beta=1$ 进行反射得到反射点 v_f，原始的简单单纯形要求将 v_f 与原单纯形中剩余的点进行比较，找出新的最坏点 v_w 继续寻优。如反射点在新的单纯形中本身就是最坏点，则不返回反射（这将形成死循环），而是将新单纯形中的次坏点去掉，按前面弃去最坏点相同的方法反射（取 $\beta=1$）。如反射点超出实验条件可行域范围，即无法按其坐标所示条件安排试验，将此点赋予很差的响应值，即令其为当前最差点。如某一试验点连续在 $s+1$ 个单纯形中是保留点，则应予重复，考察该点是否确为当前最佳点，或是误差引起。如确认该点是当前最佳点，保留该点缩小步长继续进行单纯形寻优。

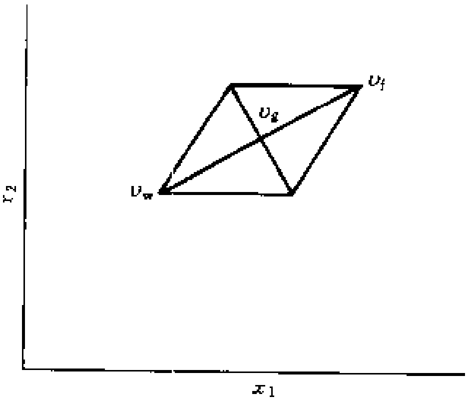


图 4-11　单纯形的反射

改进的单纯形法能加速最优条件的寻找。考虑到反射前 v_w 是最坏点，可能出现以下三种情况。

第一种情况，反射点 v_r 是当前最佳点，即较原试验点中最佳点更优。这时自然考虑是否继续沿由 v_w 反射到 v_r 的方向延伸。这就是说，当 v_r 是当前最佳点，进行延伸到 v_e

$$v_e = v_g + \gamma(v_r - v_w) \tag{4-67}$$

在计算延伸点 v_e 的式（4-67）中，$\gamma>1$，称延伸系数（扩大系数），通常取 $\gamma=2$。如得到的 v_e 点优于点 v_r，保留 v_e 与原留下的各点（不含 v_r、v_w）构成新单纯形（已不是正规单纯形），继续寻优。

第二种情况，v_r 不是当前最佳点，但亦非除 v_w 以外的最差点（除 v_w 以外，v_r 还比 v_i 中某点好），这时保留 v_r，弃去 v_w，组成新单纯形继续寻优。

第三种情况，v_r 较原单纯形次差点差，即暂不计 v_w，v_r 是最差点。这时考虑收缩，需注意：

① v_r 比 v_w 好，舍去 v_w，对 $v_r - v_g$ 收缩为 v_c，即 $v_c = v_g + \gamma(v_r - v_g)$

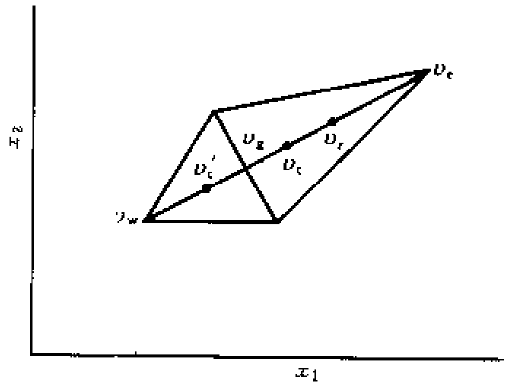


图 4-12　单纯形的反射、延伸与收缩

在此 γ 是收缩系数，一般取 $\gamma=1/2$。如 v_c 比 v_r 好，保留 v_c，与除 v_w 以外的原顶点组成单纯形再寻优；如 v_c 比 v_r 差，舍去 v_c，缩小原单纯形继续寻优。所谓缩小单纯形，是指保持原最佳点，将棱长缩小（例如减半）形成单纯形。

② v_r 比 v_w 差，舍去 v_r，对 $v_w - v_g$ 收缩为 v_c'，即 $v_c' = v_g + \gamma(v_w - v_g)$

将 v_c 与 v_w 比较，如 v_c 比 v_w 好，保留 v_c'，与除 v_w 以外的原顶点组成单纯形再寻优；如 v_c' 比 v_w 差，舍去 v_c'，保留原最佳点将棱长缩小的单纯形继续寻优。

在 $s=2$ 的情况下的反射、延伸与收缩的情况示意于图 4-12。

改进单纯形法的终止准则仍与前述简单单纯形法相同。在一个点经过 $n+1$ 次单纯形移动后仍未被淘汰，即应予复核，如被肯定是当前最优点，缩小步长继续试验，直至继续缩小步长已达到精度为止。单纯形试验寻优法一般说来只能找到局部最优点，因其本质上还只是一

种局部最优方法。但因其方便易行，在化学研究中得到了很广泛的应用。

第八节　混料试验设计方法

一、基本概念

化工、材料、冶金、食品、精细化工等领域的产品大多是几种化学成分混合后加工制造出来的，它们的某项或某几项特性指标与混合成分所占比例有关。如何确定各成分在混料（或称配方）中所占比例，以使我们感兴趣的特性指标在一定条件下达到最优，是生产和科研中的一个重要问题。在这一方面国内已有一些专著出版，本节只讨论一些基本问题，读者可参阅文献［24］。在混料问题中，试验的响应值仅与每种组分的百分比有关，而与混料的总量无关。每种或分所占的百分比只能在0～1之间变化，其相加总和必须是1，设 $x_i(i=1,2,\cdots,s)$ 是第 i 种成分的百分比，则混料问题要受如下条件限制：

$$x_i\geqslant 0\quad (i=1,2,\cdots,s)$$
$$\sum x_i=1$$

因此，混料问题的试验区域是

$$\Re=\{x\,|\,x_i\geqslant 0\quad (i=1,2,\cdots,s);\sum x_i=1\}\tag{4-68}$$

由于有了 $\sum x_i=1$ 的限制，在 s 个混料变量 x_i $(i=1,2,\cdots,s)$ 中，只有 $s-1$ 个是独立的，这样的条件限制决定了在混料设计中不能采用一般的 s 元 d 次完全多项式回归模型，否则引起设计信息矩阵退化亏秩，因此，混料设计中常采用Scheffe典型多项式回归模型。例如一般的三元二次回归模型为

$$y=\beta_0+\sum_{j=1}^{3}\beta_j x_j+\sum_{k\leqslant j}^{3}\beta_{ij}x_k x_j+\sum_{k=1}^{3}\beta_{ij}x_k^2\tag{4-69}$$

而混料回归设计中的三分量二次回归方程为

$$y=\sum_{j=1}^{3}\beta_j x_j+\sum_{k\leqslant j}^{3}\beta_{ij}x_k x_j\tag{4-70}$$

它没有常数项和平方项，只有一次项和交叉项。这是因为，假定混料回归设计中的三分量二次回归方程为

$$y=\beta_0+\sum_{j=1}^{3}\beta_j x_j+\sum_{k\leqslant j}^{3}\beta_{ij}x_k x_j+\sum_{k=1}^{3}\beta_{ij}x_k^2\tag{4-71}$$

由 $x_1+x_2+x_3=1$ 可得

$$\beta_0=\beta_0x_1+\beta_0x_2+\beta_0x_3\tag{4-72}$$
$$x_1^2=x_1(1-x_2-x_3)=x_1-x_1x_2-x_1x_3\tag{4-73a}$$
$$x_2^2=x_2(1-x_1-x_3)=x_2-x_1x_2-x_2x_3\tag{4-73b}$$
$$x_3^2=x_3(1-x_1-x_2)=x_3-x_1x_3-x_2x_3\tag{4-73c}$$

将上述式（4-72）与式（4-73）代入（4-71）式，则有

$$\begin{aligned}y=&(\beta_0+\beta_1+\beta_{11})x_1+(\beta_0+\beta_2+\beta_{22})x_2+(\beta_0+\beta_3+\beta_{33})x_3+\\&(\beta_{12}-\beta_{11}-\beta_{22})x_1x_2+(\beta_{13}-\beta_{11}-\beta_{33})x_1x_3+(\beta_{23}-\beta_{22}-\beta_{33})x_2x_3\end{aligned}\tag{4-74}$$

这实际就是Scheffe典型多项式回归模型［式（4-70）］。

对于 s 分量的混料回归设计，其Scheffe典型多项式回归模型分别为

一次式

$$y=\sum_{j=1}^{s}\beta_j x_j\tag{4-75a}$$

二次式

$$y = \sum_{j=1}^{s}\beta_j x_j + \sum_{k\leqslant j}\beta_{kj}x_k x_j \tag{4-75b}$$

三次式

$$y = \sum_{j=1}^{s}\beta_j x_j + \sum_{k\leqslant j}\beta_{kj}x_k x_j + \sum_{k\leqslant j}\gamma_{kj}x_k x_j(x_k - x_j) + \sum_{k\leqslant j\leqslant l}\beta_{kjl}x_k x_j x_l \tag{4-75c}$$

在后面将可以看到，当使用单纯形格子点设计时，Scheffe 典型多项式回归模型中回归系数的最小二乘估计的计算将变得很简单。

混料模型区域的单纯形表示法 在 s 维试验空间中，s 因子的混料区域 $\mathfrak{R}$ 可用 $(s-1)$ 维单纯形来表示。单纯形上的点，如其 s 个坐标中有 1 个坐标为 1，其余 $(s-1)$ 个坐标为 0，则这样的点称为单纯形顶点。因此，在 s 因素的混料模型中，单纯形的顶点有 s 个，下以 $s=3$ 为例来讨论单纯形上点的坐标问题。

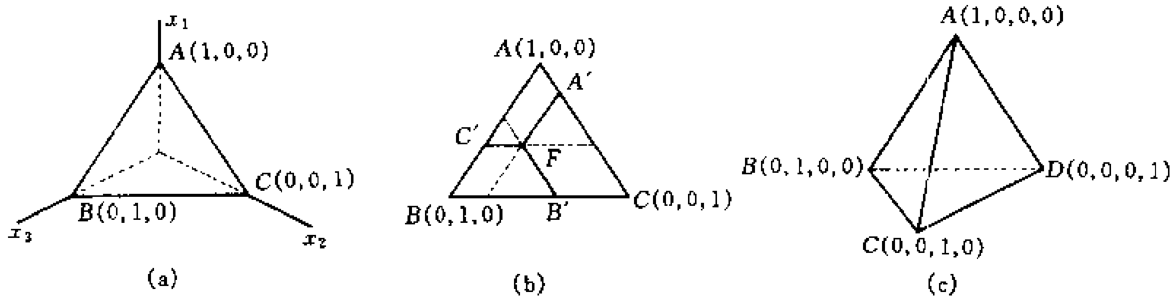
当 $s=3$ 时，其单纯形为平面 $x_1+x_2+x_3=1$ 上的一个等边三角形 [参见图 4-13 (a)]，它的顶点为 $A(1,0,0)$，$B(0,1,0)$，$C(0,0,1)$。在 $x_1+x_2+x_3=1$ 的条件限制下，各分量 x_i $(i=1,2,3)$，只能在三角形 ABC 上取值。为了方便，在使用时不再画出 3 个坐标轴，只画出一个等边三角形就可以了 [参见图 4-13 (b)]。对于该三角形中的任一点 F 有

$$FA'+FB'+FC'=1 \tag{4-76}$$

这样，就可以把 FA' 的长度看成是 F 点的 x_1 坐标值，把 FB' 和 FC' 的长度分别看成是 F 点的 x_2 和 x_3 坐标值，即单纯形上的点与有序数 (x_1, x_2, x_3) 之间建立了一一对应关系，显然，x_1，x_2，x_3 满足混料条件。

四因素混料设计的单纯形是一个四面体 [参见图 4-13 (c)]，它的 4 个顶点分别为 $A(1,0,0,0)$，$B(0,1,0,0)$，$C(0,0,1,0)$，$D(0,0,0,1)$，与 $s=3$ 时类似，可用有序数 x_1,x_2,x_3,x_4 分别表示该单纯形内的点的坐标，以建立三维正规单纯形坐标系。

用类似的方法，可建立 s 个因素空间的混料设计的正规单纯形坐标系。



(a)　　(b)　　(c)

图 4-13　混料试验区域

二、单纯形格子点

对于 Scheffe 典型多项式回归模型，Scheffe 提出了单纯形格子设计方法，这种设计方法可以保证设计点分布均匀，且回归系数的计算变得简单。下面先介绍格子点的概念和计算公式。

前已述及，当 $s=3$ 时，其单纯形是一个边长为 1 的等边三角形，它的 3 个顶点的全体称为一阶格子点集，记为 $\{3,1\}$[参见图 4-14(a)]。

将等边三角形的三边各二等分，则此三角形的三个顶点与三条边的中点 [图 4-14 (b)] 的全体称为二阶格子点集，记为 $\{3,2\}$，其中共有 6 个点，各点坐标参见表 4-37。

将等边三角形的三边三等分，对应的分点连成与边平行的直线，参见图 4-14 (c)，在等

边三角形上形成许多格子，这些格子的顶点的全体称为三阶格子点集，记为{3,3}，其中有 10 个点，各点坐标示于表 4-38。

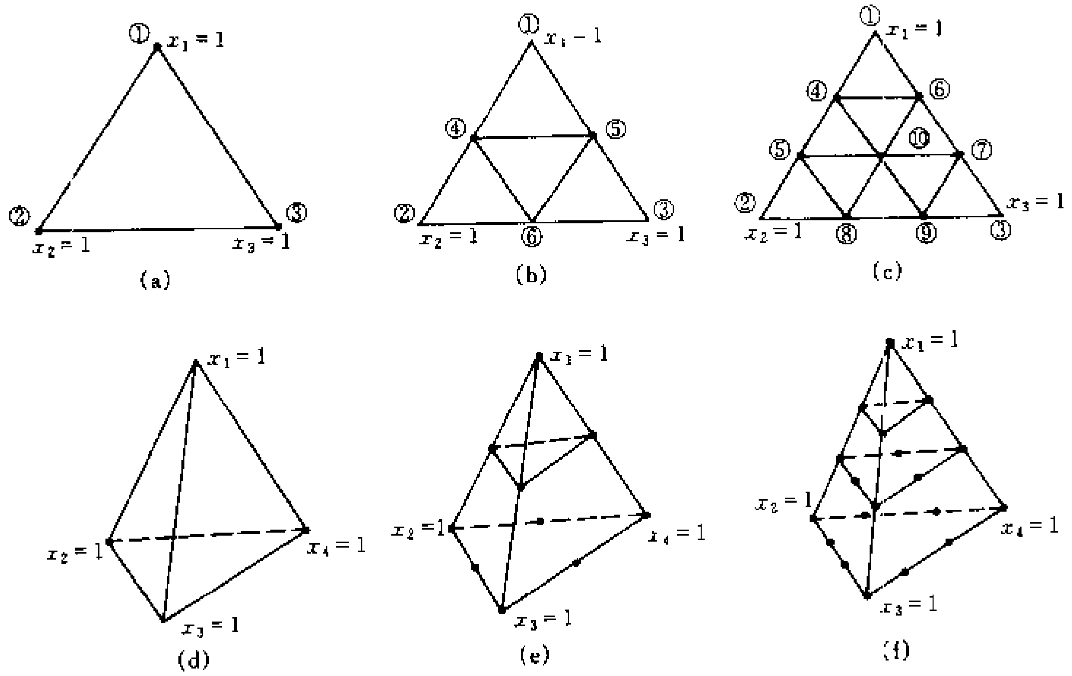


图 4-14　单纯形格子设计点分布

表 4-37　单纯形格子 {3, 2} 各点坐标

点号＼坐标	x_1	x_2	x_3	点号＼坐标	x_1	x_2	x_3
1	1	0	0	4	1/2	1/2	0
2	0	1	0	5	1/2	0	1/2
3	0	0	1	6	0	1/2	1/2

表 4-38　单纯形格子 {3, 3} 各点坐标

点号＼坐标	x_1	x_2	x_3	点号＼坐标	x_1	x_2	x_3
1	1	0	0	6	2/3	0	1/3
2	0	1	0	7	1/3	0	2/3
3	0	0	1	8	0	2/3	1/3
4	2/3	1/3	0	9	0	1/3	2/3
5	1/3	2/3	0	10	1/3	1/3	1/3

将等边三角形各边 d 等分，类似地可得 d 阶格子点集，记为{3,d}。

当 $s=4$ 时，类似地可得 d 阶格子点集，记为{4,d}，格子点集{4,1}，{4,2}，{4,3}，如图 4-14 中的 (d)、(e)、(f) 所示，格子点集{4,2} 中有 10 个点，各点坐标见表 4-39，格子点集{4,3} 中有 20 个点，各点坐标见表 4-40。

表 4-39　单纯形格子 {4, 2} 各点坐标

点号＼坐标	x_1	x_2	x_3	x_4	点号＼坐标	x_1	x_2	x_3	x_4
1	1	0	0	0	6	1/2	0	1/2	0
2	0	1	0	0	7	1/2	0	0	1/2
3	0	0	1	0	8	0	1/2	1/2	0
4	0	0	0	1	9	0	1/2	0	1/2
5	1/2	1/2	0	0	10	0	0	1/2	1/2

表 4-40 单纯形格子〈4，3〉各点坐标

点号＼坐标	x_1	x_2	x_3	x_4	点号＼坐标	x_1	x_2	x_3	x_4
1	1	0	0	0	11	0	2/3	1/3	0
2	0	1	0	0	12	0	1/3	2/3	0
3	0	0	1	0	13	0	2/3	0	1/3
4	0	0	0	1	14	0	1/3	0	2/3
5	2/3	1/3	0	0	15	0	0	2/3	1/3
6	1/3	2/3	0	0	16	0	0	1/3	2/3
7	2/3	0	1/3	0	17	1/3	1/3	1/3	0
8	1/3	0	2/3	0	18	1/3	1/3	0	1/3
9	2/3	0	0	1/3	19	1/3	0	1/3	1/3
10	1/3	0	0	2/3	20	0	1/3	1/3	1/3

将上述作法推广，可做出其他的格子点集，格子点集一般地可表示为$\{s,d\}$，其中 s 表示单纯形顶点的个数，d 表示将单纯形的边长等分的份数。

下面介绍（$s-1$）维正规单纯形（有 s 个顶点）d 阶格子点集$\{s,d\}$ 中各格子点的正规单纯形坐标的一般算法，取 s 个互相正交的单位矢量

$$\boldsymbol{a}_1:(1,\ 0,\ 0,\ \cdots,\ 0)$$
$$\boldsymbol{a}_2:(0,\ 1,\ 0,\ \cdots,\ 0)$$
$$\vdots \qquad\qquad \vdots$$
$$\boldsymbol{a}_s:(0,\ 0,\ \cdots,\ 1)$$

则这 s 个单位矢量作为顶点围成一个正规单纯形，此单纯形上的任一点 $\boldsymbol{x}$ 都可以表示为

$$\boldsymbol{x}=i_1\boldsymbol{a}_1+i_2\boldsymbol{a}_2+\cdots+i_s\boldsymbol{a}_s \tag{4-77}$$

式中

$$i_1,i_2,\cdots,i_s\geqslant 0;\quad i_1+i_2+\cdots+i_s=1 \tag{4-78}$$

当 i_1，i_2，…，i_s 都取分母是 d 的分数时，即

$$i_1=\alpha_1/d,i_2=\alpha_2/d,\cdots,i_s=\alpha_s/d$$
$$\alpha_i\geqslant 0(i=1,2,\cdots,s)$$
$$\alpha_1+\alpha_2+\cdots+\alpha_s=d$$

则由式（4-78）所确定的总体就是（$s-1$）维正规单纯形的 d 阶格子点集$\{s,d\}$，也就是说

$$\{s,d\}=\left\{(\alpha_1/d,\alpha_2/d,\cdots,\alpha_s/d)\ \middle|\ \begin{matrix}\alpha_i\geqslant 0(i=1,2,\cdots,s)\\ \alpha_1+\alpha_2+\cdots+\alpha_s=d\end{matrix}\right\} \tag{4-79}$$

即由式（4-79）所确定的点集实际有 C_{s+d-1}^{d}个点，利用上式可算出$\{s,d\}$ 中各点的单纯形坐标系的坐标。下面以 $s=4$ 的情况为例，算出$\{4,2\}$，$\{4,3\}$ 各点的坐标。

（1）$s=4$，$d=2$，即 4 维 2 阶格子点集$\{4,2\}$ 单纯形坐标系的坐标算法：

因 $d=2$，$\alpha_i(i=1,2,\cdots,s)$的取值情况只有两种，即 1 或 2。

① 当某个 α_i 取 2 时，其余的 $\alpha_j(j=1,2,\cdots,4,j\neq i)$都得为零，则此时有 $C_4^1=4$ 个点；

② 当某两个 α_i 取 1 时，其余的都得为零，则此时有 $C_4^2=6$ 个点。

由此算得的 4 维 2 阶格子点集$\{4,2\}$，10 个单纯形坐标系的坐标如表 3-39 所示。

（2）$s=4$，$d=3$，即 4 维 3 阶格子点集$\{4,3\}$ 单纯形坐标系的坐标算法：

因 $d=3$，$\alpha_i(i=1,\ 2,\ \cdots,\ s)$ 的取值情况有 3 种，即 1、2 或 3。

① 当某个 α_i 取 3 时，其余的 $\alpha_j(j=1,\ 2,\ \cdots,\ 4,\ j\neq i)$ 都得为零，则此时有 $C_4^1=4$ 个点；

② 当某个 α_i 取 2，另一个取 1 时，其余的 α_i 都得为零，则此时有 $2C_4^2=12$ 个点；

③ 当某三个 α_i 取 1 时，剩余的为零，则此时有 $C_4^3=4$ 个点。

由此算得的 4 维 3 阶格子点集 $\{4,3\}$ 的 20 个单纯形坐标系的坐标如表 4-40 所示。

三、单纯形格子设计法

设有 s 分量系统，各分量 x_i（$i=1, 2, \cdots, s$）的变化范围满足约束条件 $\sum x_i=1$，当采用 d 阶典型的 Scheffe 多项式回归模型时，试验点选为 $\{s, d\}$ 的 C_{s+d-1}^{d} 个格子点，这正好与所采用的 d 阶典型的 Scheffe 多项式回归方程中，待估计的回归系数的个数相等，故单纯形格子设计是饱和设计，是在“试验次数最少”意义下的最优设计。常用的单纯形格子设计的试验点数及相应的典型的 Scheffe 多项式回归方程阶数 d 之间的关系如表 4-41 所示。

表 4-41 单纯形格子设计的试验点数

分量数 \ 点数 \ d	回归方程阶数			分量数 \ 点数 \ d	回归方程阶数		
	2	3	4		2	3	4
3	6	10	15	6	21	56	126
4	10	20	35	8	36	120	330
5	15	35	70	10	55	220	715

前面说过，对于典型的 Scheffe 多项式回归模型，当采用单纯形格子点设计时，回归系数的最小二乘估计将变得很简单，这时，每个回归系数的值只取决于按一定规律对应的一些格子点上的观察值，而与其他设计点上的观察值无关，各回归系数都可以表示成相应设计点上观察值的简单线性组合。下面以分量混料设计的二阶多项式回归方程为例，说明怎样从试验结果计算各回归系数。

二阶多项式回归方程典型 Scheffe 形式为

$$y=\beta_1x_1+\beta_2x_2+\beta_3x_3+\beta_{12}x_1x_2+\beta_{13}x_1x_3+\beta_{23}x_2x_3 \tag{4-80}$$

与此相应，单纯形格子设计及试验结果见表 4-42。

表 4-42 单纯形格子 {3, 2} 设计及试验结果

试验点	x_1	x_2	x_3	试验结果	试验点	x_1	x_2	x_3	试验结果
1	1	0	0	y_1	4	1/2	1/2	0	y_{12}
2	0	1	0	y_2	5	1/2	0	1/2	y_{13}
3	0	0	1	y_3	6	0	1/2	1/2	y_{23}

为表述清晰起见，各试验点的观察值以相应的下标来表示，例如：

y_i——x_i 为 1 而其余分量皆为零的格子点的观察值；

y_{ij}——x_i 为 1/2，x_j 为 1/2，其余分量皆为零的格子点的观察值；

y_{iij}——x_i 为 2/3，x_j 为 1/3，其余分量皆为零的格子点的观察值；

y_{iiij}——x_i 为 3/4，x_j 为 1/4，其余分量皆为零的格子点的观察值，等等。

对于式（4-80）表示的回归模型，可容易地看出

$$\beta_1=y_1$$

$$\beta_{12}=4y_{12}-2(y_1+y_2)$$

写成一般式，有

$$\beta_i=y_i$$

$$\beta_{ij}=4y_{ij}-2(y_i+y_j) \quad (i<j; i,j=1,2,3)$$

对于一般 s 分量模型，与二阶单纯形格子设计相应的典型 Scheffe 多项式回归方程的系数公式为

$$\beta_i = y_i$$

$$\beta_{ij} = 4y_{ij} - 2(y_i + y_j) \quad (i<j; i,j=1,2,\cdots,s)$$

类似地，s 分量三阶单纯形格子设计相应的典型 Scheffe 多项式回归方程的系数公式为

$$\beta_i = y_i$$

$$\beta_{ij} = 4/9(y_{iij} + y_{ijj} - y_i - y_j) \quad (i<j; i,j=1,2,\cdots,s)$$

$$\gamma_{ij} = 4/9(3y_{iij} - 3y_{ijj} - y_i + y_j)$$

$$\beta_{kji} = 273y_{ijk} - 27/4(y_{iij} + y_{ijj} + y_{iik} + y_{ikk} + y_{jjk} + y_{jkk}) + 9/2(y_i + y_j + y_k)$$

$$(i<j<k; i,j,k=1,2,\cdots,s)$$

四、单纯形重心设计法

在单纯形格子设计中，当所采用的回归模型的阶数大于 2 时，在某些混料中，各分量是以不相等的比例出现的。能否对单纯形格子设计进行改进，只考虑各成分有等比例成分的试验？另外，单纯形格子设计虽然是饱和设计，但试验次数仍然很多（C_{s+d-1}^{d}）。这样，要求对 Scheffe 典型多项式回归模型进行改造，改变为对各分量来说都是对称的。于是，提出了单纯形重心设计。

在单纯形重心设计中采用的回归模型为

一次重心多项式

$$y = \sum_{j=1}^{s} \beta_j x_j \tag{4-81a}$$

二次重心多项式

$$y = \sum_{j=1}^{s} \beta_j x_j + \sum_{k<j} \beta_{kj} x_k x_j \tag{4-81b}$$

三次重心多项式

$$y = \sum_{j=1}^{s} \beta_j x_j + \sum_{k<j} \beta_{kj} x_k x_j + \sum_{k<j<i} \beta_{kji} x_k x_j x_i \tag{4-81c}$$

由上述重心多项式回归方程的形式可以看出，一阶、二阶重心多项式回归方程与一阶、二阶 Scheffe 典型多项式回归方程是相同的。当阶数大于 2 时，前者较后者减少了一些乘积项，另外，对于 s 分量混料系统，重心多项式回归方程的阶数最高为 s。

在单纯形重心设计中，试验点的选取方法是取单纯形顶点及它们的一些重心点，如

单纯形的 s 个顶点：(1,0,0,…,0)，…，(0,0,…,1)，共有 C_s^1 个点。

两个顶点的重心点：(1/2,1/2,0,…,0)，…，(0,0,…,1/2,1/2)，共有 C_s^2 个点。

三个顶点的重心点：(1/3,1/3,1/3,0,…,0)，(1/3,1/3,0,1/3,…,0)，…，(0,0,…,1/3,1/3,1/3)，共有 C_s^3 个点。

($s-1$) 个顶点的重心点：(1/($s-1$),1/($s-1$),1/($s-1$),…,1/($s-1$),0)，…，(0,1/($s-1$),1/($s-1$),1/($s-1$),…,1/($s-1$))，共有 $C_s^{s-1}=C_s^1$ 个点。

s 个顶点的重心点：(1/s,1/s,1/s,…,1/s)，共有 $C_s^s=1$ 个点。

对 $s=3$，其单纯形重心设计共有 7 个点，见表 4-43 及图 4-15。

表 4-43　三分量三阶单纯形重心设计

试验点	x_1	x_2	x_3	试验结果
1	1	0	0	y_1
2	0	1	0	y_2
3	0	0	1	y_3
4	1/2	1/2	0	y_{12}
5	1/2	0	1/2	y_{13}
6	0	1/2	1/2	y_{23}
7	1/3	1/3	1/3	y_{123}

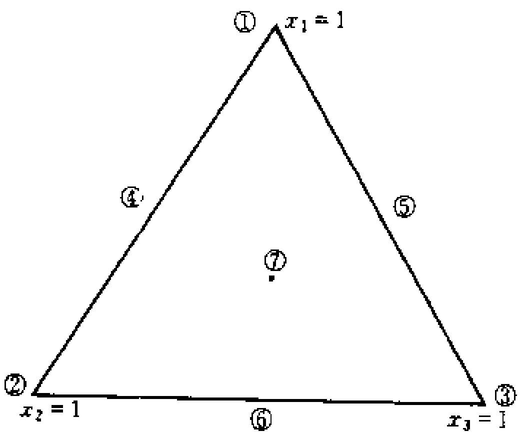


图 4-15　单纯形重心设计示意图

对 $s=4$，其单纯形重心设计共有 15 个点，见表 4-44，类似地可以写出 $s>4$ 的各阶单纯形重心设计方案。

容易看出，对 s 因子混料设计系统，前面所说的单纯形顶点及顶点的各种重心点总数为

$$C_s^1+C_s^2+C_s^3+\cdots+C_s^s=2^s-1 \tag{4-82}$$

当采用 s 阶重心多项式回归模型时，应取这全部 2^s-1 个点作为试验点。显然，单纯形重心设计方案的试验点数等于相应的 s 阶重心多项式回归方程中待估系数的个数，故单纯形重心设计也是饱和设计。一、二阶单纯形格子设计与一、二阶单纯形重心设计相同，当 $d>2$ 时，对于同阶多项式回归模型来说，单纯形重心设计的试验点数要少于单纯形格子设计的试验点数，且 d 越大其差越大。

表 4-44　四分量四阶单纯形重心设计

试验点	x_1	x_2	x_3	x_4	响应值	试验点	x_1	x_2	x_3	x_4	响应值
1	1	0	0	0	y_1	9	0	1/2	0	1/2	y_1
2	0	1	0	0	y_2	10	0	0	1/2	1/2	y_2
3	0	0	1	0	y_3	11	1/3	1/3	1/3	0	y_3
4	0	0	0	1	y_4	12	1/3	1/3	0	1/3	y_4
5	1/2	1/2	0	0	y_1	13	1/3	0	1/3	1/3	y_1
6	1/2	0	1/2	0	y_2	14	0	1/3	1/3	1/3	y_2
7	1/2	0	0	1/2	y_3	15	1/4	1/4	1/4	1/4	y_3
8	0	1/2	1/2	0	y_4						

对混料设计，也可以考虑 D-最优性问题，可以证明，一阶、二阶单纯形格子设计及三阶单纯形重心设计是 D-最优的，而三阶单纯形格子设计不是 D-最优的。

三分量三阶重心多项式回归方程

$$y=\sum_{i=1}^{3}\beta_i x_i+\sum_{i<j}\beta_{ij}x_i x_j+\beta_{123}x_1x_2x_3 \tag{4-83}$$

的系数计算公式为

$$\beta_i=y_i$$

$$\beta_{ij}=4y_{ij}-2(y_i+y_j)\quad (i<j;i,j=1,2,3)$$

$$\beta_{123}=27y_{123}-12(y_{12}+y_{13}+y_{23})+3(y_1+y_2+y_3)$$

一般地，对于 s 分量混料系统，采用 s 阶单纯形重心试验设计方案时，相应的 d 阶重心多项式回归方程各回归系数按式（4-84）计算：

$$\beta_{Dr}=r\sum_{1\leqslant t\leqslant r}(-1)^{r-t}t^{r-1}y_t(Dr)\qquad(r=1,2,\cdots,s)\tag{4-84}$$

式中 Dr——s 个成分中某 r 个的集合；

$y_t(Dr)$——从这 s 个成分中取 t 个的全部 C_r^t 个组合的试验值的总和。

例如，$s=4$，$d=4$ 时，四阶重心多项式回归方程为

$$y=\sum_{i=1}^{4}\beta_i x_i+\sum_{i\leqslant j}\beta_{ij}x_ix_j+\sum_{i\leqslant j\leqslant k}\beta_{ijk}x_ix_jx_k+\beta_{1234}x_1x_2x_3x_4$$

当采用四阶单纯形重心设计方案时，上式中各回归系数计算如下：

令 $r=1$，则 $t=1$，从式（4-84）得

$$\beta_i=1[(-1)^{1-1}1^{1-1}y_i]=y_i\qquad(i=1,2,3,4)$$

令 $r=2$，则 $t=1$、2，从式（4-84）得

$$\begin{aligned}\beta_{ij}&=2[(-1)^{2-2}\times2^{2-1}y_{ij}+(-1)^{2-1}\times1^{2-1}(y_i+y_j)]\\&=4y_{ij}-2(y_i+y_j)\quad(i,j=1,2,3,4;i<j)\end{aligned}$$

令 $r=3$，则 $t=1$、2、3，从式（4-84）得

$$\begin{aligned}\beta_{ijk}&=3[(-1)^{3-3}\times3^{3-1}y_{ijk}+(-1)^{3-2}\times2^{3-1}(y_{ij}+y_{ik}+y_{jk})+(-1)^{3-1}\times1^{3-1}(y_i+y_j+y_k)]\\&=27y_{ijk}-12(y_{ij}+y_{ik}+y_{jk})+3(y_i+y_j+y_k)\quad(i,j,k=1,2,3,4;i<j<k)\end{aligned}$$

令 $r=4$，则 $t=1$、2、3、4，从式（4-84）得

$$\begin{aligned}\beta_{1234}&=4[(-1)^{4-4}\times4^{4-1}y_{1234}+(-1)^{4-3}\times3^{4-1}(y_{123}+y_{124}+y_{134}+y_{234})\\&\quad+(-1)^{4-2}\times2^{4-1}(y_{12}+y_{13}+y_{14}+y_{23}+y_{24}+y_{34})+(-1)^{4-1}\times1^{4-1}(y_1+y_2+y_3+y_4)]\\&=256y_{1234}-108(y_{123}+y_{124}+y_{134}+y_{234})+32(y_{12}+y_{13}+y_{14}+y_{23}+y_{24}+y_{34})\\&\quad-4(y_1+y_2+y_3+y_4)\end{aligned}$$

以上论及的混料设计方法只能说是最基本的设计方法，实际上，以前讨论的均匀设计，也可构造出混料设计的方案来，读者可参阅文献［25］。

第九节　化学中常用的优化方法

一、基本概念简介

在化学研究和分析化学中，需经常使用最优化方法，前述的单纯形试验设计方法本身就是一种最优化方法，它不但可用于试验设计，还可用于计算寻优，即用于多元校正、多元分辨、模式识别、化学专家系统及化学动力学参数辨识等领域，可看成是一种通用的数学工具[26~28]。所以，在此对化学中常用的优化方法作简要的专门介绍。

在介绍化学中常用优化方法之前，先讨论有关最优化理论的一些基本概念。

1. 优化问题迭代算法的基本思想

在优化问题求解中，常须用到迭代方法。迭代方法直接从分析目标函数 $f(\boldsymbol{x})$ 的特征入手，构造一类逐次使目标函数值下降的搜索。因一般在化学研究中使用的优化方法多由迭代法构成，对这类方法的基本思路的掌握将对理解具体算法带来很大好处，所以，将对此也作简要介绍。

这类方法一般由以下四个步骤组成：

① 选择初始近似点 $\boldsymbol{x}^{(0)}$（自然是越靠近最优点越好）。

② 如已算出第 k 次近似点 $\boldsymbol{x}^{(k)}$，但 $\boldsymbol{x}^{(k)}$还不是在要求的误差范围内的最优解，此时可选择一个搜索方向 $\boldsymbol{s}^{(k)}$，使沿 $\boldsymbol{s}^{(k)}$方向目标函数值下降（设最优是最小），即 $\boldsymbol{s}^{(k)}$应是 $f(\boldsymbol{x})$的下降方向。

③ 由点 $\boldsymbol{x}^{(k)}$出发，沿 $\boldsymbol{s}^{(k)}$方向作射线，在此射线上，定出步长 λ_k，使所得的点

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\lambda_k\boldsymbol{s}^{(k)}$$

满足

$$f(\boldsymbol{x}^{(k+1)})<f(\boldsymbol{x}^{(k)})$$

在多数算法中，λ_k 的选取应使 $f(\boldsymbol{x})$的下降得最多，即从 $\boldsymbol{x}^{(k)}$出发，沿 $\boldsymbol{s}^{(k)}$方向求 $f(\boldsymbol{x})$的极小值，这是单变量 λ 的函数极小化问题，称为一维搜索，其表达式为

$$f(\boldsymbol{x}^{(k+1)})=f(\boldsymbol{x}^{(k)}+\lambda_k\boldsymbol{s}^{(k)})=\min_{\lambda}f(\boldsymbol{x}^{(k)}+\lambda\boldsymbol{s}^{(k)}) \tag{4-85}$$

④ 验证所得新点 $\boldsymbol{x}^{(k+1)}$是否达到要求的精度 ε（ε 为很小的正数），即

$$\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}\|\leqslant\varepsilon$$

或满足其他类型的终止标准。若满足，$\boldsymbol{x}^{(k+1)}$就可以作为最优点，否则，以 $\boldsymbol{x}^{(k+1)}$作为新的初始点，转②继续进行搜索。

由上述计算过程可以看出，在这样的迭代算法中，搜索方向 $\boldsymbol{s}^{(k)}$和步长因子 λ_k 构成每一次迭代的修正量，它们是决定算法好坏的最重要因素。

2. 局部最优与全局最优

记 $f(\boldsymbol{x})$是定义在 n 维欧氏空间 R^n 上的实函数，那么，所谓无约束极值（优化）问题，是要求在 R^n 上求函数 $f(\boldsymbol{x})$的最小值（或最大值），即

$$\min_{\boldsymbol{x}\in R^n}f(\boldsymbol{x})$$

所谓约束极值问题，是求解

$$\min_{\boldsymbol{x}\in E^n}f(\boldsymbol{x})$$

其中，$E^n=\{\boldsymbol{x}\,|\,g_i(x)\leqslant 0, i=1,2,3,\cdots,m\}\subset R^n$。$g_i(x)\leqslant 0$，$(i=1,2,3,\cdots,m)$为该问题的约束条件。

对于点 $\boldsymbol{x}^*\in R^n$，如果不等式 $f(\boldsymbol{x}^*)\leqslant f(\boldsymbol{x})$对所有的 $\boldsymbol{x}\in R^n$ 都成立，则称点 $\boldsymbol{x}^*$ 为最优化问题的全局最优解。

如果存在 $\boldsymbol{x}^*$ 的某个邻域 $V(\boldsymbol{x}^*)$，如果不等式 $f(\boldsymbol{x}^*)\leqslant f(\boldsymbol{x})$对所有的 $\boldsymbol{x}\in V(\boldsymbol{x}^*)$都成立，则称点 $\boldsymbol{x}^*$ 为最优化问题的局部最优解。

3. 函数的梯度及二阶导数矩阵

函数 $f(\boldsymbol{x})$在给定点的梯度是一向量，它的大小就是函数在该点的方向导数的最大值，它的方向垂直于函数过该点的等值面，且指向函数增大的方向。一般说来，其梯度向量可表示为

$$\nabla f(\boldsymbol{x})=[\partial f(\boldsymbol{x})/\partial x_1,\partial f(\boldsymbol{x})/\partial x_2,\cdots,\partial f(\boldsymbol{x})/\partial x_n]^{t} \tag{4-86}$$

其二阶导数矩阵定义为

$$\nabla^2 f(\boldsymbol{x})=\begin{bmatrix}\partial^2 f(\boldsymbol{x})/\partial x_1^2 & \partial^2 f(\boldsymbol{x})/\partial x_1\partial x_2 & \cdots & \partial^2 f(\boldsymbol{x})/\partial x_1\partial x_n\\ \partial^2 f(\boldsymbol{x})/\partial x_2\partial x_1 & \partial^2 f(\boldsymbol{x})/\partial x_2^2 & \cdots & \partial^2 f(\boldsymbol{x})/\partial x_1\partial x_n\\ \vdots & \vdots & & \vdots\\ \partial^2 f(\boldsymbol{x})/\partial x_n\partial x_1 & \partial^2 f(\boldsymbol{x})/\partial x_n\partial x_2 & \cdots & \partial^2 f(\boldsymbol{x})/\partial x_n^2\end{bmatrix}$$

二阶导数矩阵又称 Hessian 矩阵，是 $(n\times n)$ 对称矩阵。它的正定性（参见第十一章）是判定函数是否凸性的充要条件。如函数是凸的，则其最优解一定是全局最优解。

4. 共轭方向

在局部优化算法的概念起着重要的作用时，一些比较有效的方法，都是以共轭方向为搜索方向而形成的。

设 $\boldsymbol{A}$ 为一个 $(n\times n)$ 对称正定矩阵，$\boldsymbol{p}$、$\boldsymbol{q}$ 为两个 n 维向量，若

$$\boldsymbol{p}^t\boldsymbol{A}\boldsymbol{q}=0$$

则称 $\boldsymbol{p}$、$\boldsymbol{q}$ 为两个共轭向量，如果 $\boldsymbol{A}$ 为一个单位矩阵，则 $\boldsymbol{p}$、$\boldsymbol{q}$ 两向量正交。

对于正定二次 n 元函数，如果依次沿 n 个相互共轭的方向进行一维搜索，则最后得到的就是它的最优点。如果一个算法的搜索方向是互相共轭的，则称为共轭方向法，以后讨论的共轭梯度法，DFP 法都是共轭方向法。

二、局部优化算法

本节讨论的方法都是局部最优化方法，所得的最优解只是局部最优解。局部最优化方法中如按是否需要对目标函数求导来分，可分成两类，即一类需要对目标函数 $f(\boldsymbol{x})$ 进行求导运算，这样的方法包括梯度法、牛顿法、共轭梯度法以及变尺度法，这类方法充分利用了函数的解析性质，它们的收敛速度多数比不用求导的优化方法（直接法）要快；而另一类则勿需求导，常称直接法，它们包括上节讨论的单纯形法及本节将讨论的坐标轮换法和 Powell 方法。以下我们将分别对这两类局部优化算法进行简要介绍。

（一）梯度法

1. 基本思路

梯度法（或称最速下降法）是求解优化问题的一种最古老、最基本的算法。其主要思路是取函数的负梯度方向（最速下降方向）作为迭代的搜索方向，即

$$\boldsymbol{s}^{(k)}=-\nabla f(\boldsymbol{x}^{(k)})$$

至于步长的选取，可以有不同的方法，一种方法是任意取一个 λ_k，使

$$f(\boldsymbol{x}^{(k)})-\lambda_k\nabla f(\boldsymbol{x}^{(k)})<f(\boldsymbol{x}^{(k)}) \tag{4-87}$$

成立。另一种方法是沿负梯度方向找函数的一维极小点，即求解一维极小化（一维搜索）问题，即求

$$f(\boldsymbol{x}^{(k)})-\lambda_k\nabla f(\boldsymbol{x}^{(k)})=\min_{\lambda} f(\boldsymbol{x}^{(k+1)})-\lambda\nabla f(\boldsymbol{x}^{(k)}) \tag{4-88}$$

一般采用后一种方法［式（4-88）］。这种寻优方法的基本寻优思路可由图 4-16 示意表出。

2. 基本算法

① 选择初始近似点 $\boldsymbol{x}^{(0)}$ 及判别收敛的正数 ε，令 $k\Leftarrow 0$。

② 计算 $\nabla f(\boldsymbol{x}^{(k)})$

③ 若 $\|\nabla f(\boldsymbol{x}^{(k)})\|\leqslant\varepsilon_1$ 或 $\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}\|\leqslant\varepsilon_2$，则停止迭代，$\boldsymbol{x}^{(k)}$ 即为所求的解，否则进行④

④ 进行一维搜索，即求

$$f(\boldsymbol{x}^{(k)})-\lambda_k\nabla f(\boldsymbol{x}^{(k)})=\min_{\lambda} f(\boldsymbol{x}^{(k)})-\lambda\nabla f(\boldsymbol{x}^{(k)})$$

⑤ 令 $f(\boldsymbol{x}^{(k+1)})=\min f(\boldsymbol{x}^{(k)})-\lambda_k\nabla f(\boldsymbol{x}^{(k)})$，并令 $k\Leftarrow k+1$，转②。

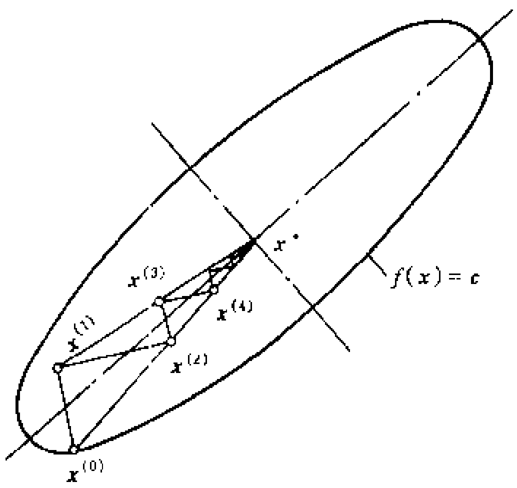


图 4-16 梯度法寻优示意图

（二）牛顿法

1. 基本思路

牛顿法也是求解优化问题的一种最古老、最基本的算法。这种算法虽然已应用不多，但它仍是应用较多的变尺度法的基础。其主要思路是对梯度法的搜索方向进行改进，即加进目标函数的二阶导数矩阵的信息，以加速收敛速度，

$$s^{(k)}=-[\nabla^2 f(x^{(k)})]^{-1}\nabla f(x^{(k)}) \tag{4-89}$$

由此可得牛顿法的迭代公式为

$$x^{(k+1)}=x^{(k)}-[\nabla^2 f(x^{(k)})]^{-1}\nabla f(x^{(k)}) \tag{4-90}$$

对于二次型目标函数，用牛顿法只须迭代一次就可得到最优点。

对于非二次函数，由于它们在极小点附近和二次函数很接近，使用牛顿法，其收敛速度也是很快的。但是牛顿法要求初始点选得比较好，如离最优点较远时，就容易发生迭代发散或收敛到非最优点的情况。为克服这一缺点，提出"阻尼牛顿法"，实际上，在牛顿法中步长因子总是取 1，而在阻尼牛顿法中，每一步迭代都是沿方向

$$s^{(k)}=-[\nabla^2 f(x^{(k)})]^{-1}\nabla f(x^{(k)})$$

的一维搜索，即以迭代公式

$$x^{(k+1)}=x^{(k)}-\lambda_k[\nabla^2 f(x^{(k)})]^{-1}\nabla f(x^{(k)})$$

代替式（4-90），其中 λ_k 使

$$f\{x^{(k)}-\lambda_k[\nabla^2 f(x^{(k)})]^{-1}\nabla f(x^{(k)})\}=\min_{\lambda} f\{x^{(k+1)}-\lambda[\nabla^2 f(x^{(k)})]^{-1}\nabla f(x^{(k)})\}$$

阻尼牛顿法保持了牛顿法收敛快的特点，而又不要求初始点很好，是有实用价值的。但阻尼牛顿法每次都要计算二阶导数矩阵的逆矩阵，当维数较高时，此工作量较大，这是它的缺点。

2. 基本算法

① 选择初始近似点 $x^{(0)}$ 及判别收敛的正数 ϵ，令 $k\Leftarrow 0$。

② 计算 $\nabla f(x^{(k)})$

③ 若 $\|\nabla f(x^{(k)})\|\leqslant\epsilon_1$ 或 $\|x^{(k+1)}-x^{(k)}\|\leqslant\epsilon_2$，则停止迭代，$x^{(k)}$ 即为所求的解，否则进行④

④ 计算 $[\nabla^2 f(x^{(k)})]^{-1}$，并令

$$s^{(k)}=-[\nabla^2 f(x^{(k)})]^{-1}\nabla f(x^{(k)})$$

⑤ 进行一维搜索，即求 λ_k

$$f(x^{(k)}+\lambda_k s^{(k)})=\min_{\lambda} f(x^{(k)}-\lambda s^{(k)})$$

⑥ 令 $x^{(k+1)}=x^{(k)}-\lambda_k[\nabla^2 f(x^{(k)})]^{-1}\nabla f(x^{(k)})$，并令 $k\Leftarrow k-1$，转②。

（三）共轭梯度法

1. 基本思路

可以证明，对于正定二次 n 元函数，如果依次沿 n 个相互共轭的方向进行一维搜索，则最后得到的就是它的最优点[27]。所谓共轭梯度法就是遵循这一思路而设计的一种优化方法。

对于一个正定二次 n 元函数

$$f(\boldsymbol{x})=(1/2)\boldsymbol{x}^{t}\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}^{t}\boldsymbol{x}+c \tag{4-91}$$

有

$$\mathrm{d}\,f(\boldsymbol{x})/\mathrm{d}\boldsymbol{x}=\nabla f(\boldsymbol{x})=\boldsymbol{A}x-\boldsymbol{b} \tag{4-92}$$

据此，可以确定优化时搜索的共轭方向。

首先任取初始点 $\boldsymbol{x}^{(0)}$，初始搜索方向为负梯度方向，即

$$\boldsymbol{s}^{(0)}=-\nabla f(\boldsymbol{x}^{(0)})$$

然后从点 $\boldsymbol{x}^{(0)}$出发，沿方向 $\boldsymbol{s}^{(0)}$进行一维搜索，得

$$f(\boldsymbol{x}^{(0)}+\lambda_0\boldsymbol{s}^{(0)})=\min_{\lambda}f(\boldsymbol{x}^{(0)}-\lambda\boldsymbol{s}^{(0)}) \tag{4-93}$$

并令

$$\boldsymbol{x}^{(1)}=\boldsymbol{x}^{(0)}-\lambda_0\boldsymbol{s}^{(0)} \tag{4-94a}$$

此时可按下述方式来构造一个与 $\boldsymbol{s}^{(0)}$共轭的方向 $\boldsymbol{s}^{(1)}$

$$\boldsymbol{s}^{(1)}=-\nabla f(\boldsymbol{x}^{(1)})+v_1\boldsymbol{s}^{(0)} \tag{4-94b}$$

这是因为与 $\boldsymbol{s}^{(0)}$共轭的方向 $\boldsymbol{s}^{(1)}$可由当前点的负梯度方向$(-\nabla f(\boldsymbol{x}^{(1)}))$与前次搜索方向 $\boldsymbol{s}^{(0)}$的线性组合而构成,问题在于如何确定 v_1。

为使 $\boldsymbol{s}^{(1)}$与 $\boldsymbol{s}^{(0)}$共轭,即需要 $\boldsymbol{s}^{(1)t}\boldsymbol{A}\boldsymbol{s}^{(0)}=0$,于是必须

$$[-\nabla f(\boldsymbol{x}^{(1)})+v_1\boldsymbol{s}^{(0)}]^{t}\boldsymbol{A}\boldsymbol{s}^{(0)}=0 \tag{4-95}$$

解方程 (4-95) 得

$$v_1=[-\nabla f(\boldsymbol{x}^{(1)})]^{t}\boldsymbol{A}\boldsymbol{s}^{(0)}/[\boldsymbol{s}^{(0)t}\boldsymbol{A}\boldsymbol{s}^{(0)}] \tag{4-96}$$

又因为

$$\nabla f(\boldsymbol{x}^{(1)})=\boldsymbol{A}\boldsymbol{x}^{(1)}+\boldsymbol{b}$$

$$\nabla f(\boldsymbol{x}^{(0)})=\boldsymbol{A}\boldsymbol{x}^{(0)}+\boldsymbol{b}$$

所以

$$\nabla f(\boldsymbol{x}^{(1)})-\nabla f(\boldsymbol{x}^{(0)})=\boldsymbol{A}(\boldsymbol{x}^{(1)}+\boldsymbol{x}^{(0)})=\lambda_0\boldsymbol{A}\boldsymbol{s}^{(0)} \tag{4-97}$$

并且因为梯度方向相互正交（见图 4-16 和图 4-17），有

$$\nabla f(\boldsymbol{x}^{(1)})^{t}\nabla f(\boldsymbol{x}^{(0)})=0$$

于是可得

$$\begin{aligned}
v_1&=[-\nabla f(\boldsymbol{x}^{(1)})]^{t}[\nabla f(\boldsymbol{x}^{(1)})-\nabla f(\boldsymbol{x}^{(0)})]/\{\boldsymbol{s}^{(0)t}[\nabla f(\boldsymbol{x}^{(1)})-\nabla f(\boldsymbol{x}^{(0)})]\}\\
&=[-\nabla f(\boldsymbol{x}^{(1)})]^{t}[\nabla f(\boldsymbol{x}^{(1)})]/\{\boldsymbol{s}^{(0)t}[\nabla f(\boldsymbol{x}^{(1)})-\nabla f(\boldsymbol{x}^{(0)})]\}\\
&=[-\nabla f(\boldsymbol{x}^{(1)})]^{t}[\nabla f(\boldsymbol{x}^{(1)})]/\{-\nabla f(\boldsymbol{x}^{(0)})^{t}[\nabla f(\boldsymbol{x}^{(1)})-\nabla f(\boldsymbol{x}^{(0)})]\}\\
&=[\nabla f(\boldsymbol{x}^{(1)})]^{t}[\nabla f(\boldsymbol{x}^{(1)})]/[\nabla f(\boldsymbol{x}^{(0)})]^{t}[\nabla f(\boldsymbol{x}^{(0)})]\\
&=\|\nabla f(\boldsymbol{x}^{(1)})\|^{2}/\|\nabla f(\boldsymbol{x}^{(0)})\|^{2}
\end{aligned}$$

同理可得

$$v_k=\|\nabla f(\boldsymbol{x}^{(k)})\|^{2}/\|\nabla f(\boldsymbol{x}^{(k-1)})\|^{2}$$

这样就可以得到一系列的共轭搜索方向 $\boldsymbol{s}^{(0)}$，$\boldsymbol{s}^{(1)}$，…，$\boldsymbol{s}^{(k)}$了。

对于一般非二次正定函数，也可仿此构造共轭搜索方向，但却不能像二次正定函数那样具有良好的有限步的收敛性。

上述共轭梯度法也称 Flether-Reeve 共轭梯度法。其基本思路示意图示于图 4-17。

2. 基本算法

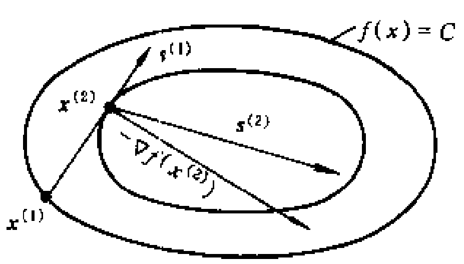


图 4-17 共轭梯度法搜索方向示意图

① 选择初始近似点 $x^{(0)}$ 及判别收敛的正数 ε，令 $k \Leftarrow 0$。

② 计算 $\nabla f(x^{(0)})$，并令 $s^{(0)}=-\nabla f(x^{(0)})$ 然后从点 $x^{(0)}$ 出发，沿方向 $s^{(0)}$ 进行一维搜索，得

$$f(x^{(0)}+\lambda_0 s^{(0)})=\min_{\lambda} f(x^{(0)}-\lambda s^{(0)})$$

并令

$$x^{(1)}=x^{(0)}-\lambda_0 s^{(0)}$$

③ 若 $\| \nabla f(x^{(k)}) \| \leqslant \epsilon_1$ 或 $\| x^{(k)}-x^{(k-1)} \| \leqslant \epsilon_2$，则停止迭代，$x^{(k)}$ 即为所求的解，否则进行④

④ 计算 $v_{k-1}=\| \nabla f(x^{(k)}) \|^2 / \| \nabla f(x^{(k-1)}) \|^2$，并令

$$s^{(k)}=-[\nabla f(x^{(k)})]+v_{k-1}s^{(k-1)}$$

⑤ 进行一维搜索，即求 λ_k

$$f(x^{(k)}+\lambda_k s^{(k)})=\min_{\lambda} f(x^{(k)}+\lambda s^{(k)})$$

⑥ 令 $x^{(k)}=x^{(k-1)}+\lambda_k s^{(k)}$，并令 $k-1 \Leftarrow k$，转③。

（四）变尺度法

1. 基本思路

变尺度法可看成是梯度法和牛顿法的一种推广，实因梯度法和牛顿法的迭代公式可统一地表示为

$$x^{(k+1)}=x^{(k)}-\lambda_k H^{(k)} \nabla f(x^{(k)}) \tag{4-98}$$

在式（4-98）中，如果 $H^{(k)}=I$，则为梯度法；若令 $H^{(k)}=[\nabla^2 f(x^{(k)})]^{-1}$，就得到阻尼牛顿法。在此，$H^{(k)}$ 为一（$n \times n$）阶矩阵，通常 $H^{(k)}\nabla f(x^{(k)})=s^{(k)}$，即 $H^{(k)}$ 可使负梯度方向旋转一个角度，所以它又被称为旋转矩阵。如前所述，梯度法收敛太慢；牛顿法要计算二阶导数矩阵和矩阵求逆，工作量大。但牛顿法收敛快，这一点是令人感兴趣的，如 $H^{(k)}$ 的选取不需要计算二阶导数矩阵和矩阵求逆，而又能逼近它，那么由式（4-98）确定的算法可能收敛得快。变尺度法的基本思路就是利用式(4-98)作为迭代公式，矩阵 $H^{(k)}$ 是在迭代过程中逐步产生的，并且它能较好地逼近 $f(x)$ 的二阶导数矩阵的逆矩阵。因这类方法的搜索方向是在 $H^{(k)}$ 度量意义下的负梯度方向，而在计算过程中，$H^{(k)}$ 又是不断变化的，故有变尺度法之称。

变尺度法有多种，目前用得较多的是 DFP 法（由 Davidon-Fletcher-Powell 提出）和 BFGS 法（由 Broyden-Fletcher-Goldfarb-Shanno 提出）。下面主要介绍 DFP 法求 $H^{(k)}$ 递推公式的基本思想。

对于一个正定二次 n 元函数，$f(x)=(1/2)x^tAx+b^tx+c$，先取初始旋转矩阵 $H^{(k)}$ 为任意一个 n 阶对称正定矩阵，例如 $H^{(1)}=I$，取初始点 $x^{(1)}$，令

$$s^{(1)}=-H^{(1)}\nabla f(x^{(1)})$$

并沿 $s^{(1)}$ 一维寻优的 $x^{(2)}$，欲求对称正定矩阵 $H^{(2)}$，可令

$$s^{(2)}=-H^{(2)}\nabla f(x^{(2)})$$

并使 $s^{(2)}$ 与 $s^{(1)}$ 为 A 共轭，因而 $H^{(2)}$ 必须满足

$$s^{(2)t}As^{(1)}=-\nabla f(x^{(2)})^t H^{(2)} A s^{(1)}=0 \tag{4-99}$$

由式（4-99）可知，如能使 $H^{(2)}As^{(1)}=s^{(1)}$，则式（4-99）必然成立。基于此，可构造 $H^{(2)}$ 的递推求法如下。

记

$$\Delta x^{(k)}=x^{(k+1)}-x^{(k)}=\lambda_k s^{(k)} \tag{4-100}$$

$$\Delta g^{(k)}=\nabla f(x^{(k+1)})-\nabla f(x^{(k)})=A(x^{(k+1)}-x^{(k)})=\lambda_k A s^{(k)} \tag{4-101}$$

式中，$k=1$，2，…，n。

设 $H^{(2)}$ 与 $H^{(1)}$有递推关系

$$H^{(2)}=H^{(1)}+\Delta H^{(1)}$$

根据 $H^{(2)}As^{(1)}=s^{(1)}$可知，未知的修正项矩阵 $\Delta H^{(1)}$应满足

$$\Delta H^{(1)}As^{(1)}=s^{(1)}-H^{(1)}As^{(1)}$$

考虑到式（4-16）和式（4-17），可将上式改写成

$$\Delta H^{(1)}As^{(1)}=\Delta H^{(1)}\Delta g^{(1)}/\lambda_k=s^{(1)}-H^{(1)}As^{(1)}=s^{(1)}(I-H^{(1)}A)=\Delta x^{(1)}/\lambda_k-H^{(1)}\Delta g^{(1)}/\lambda_k$$

即

$$\Delta H^{(1)}\Delta g^{(1)}=\Delta x^{(1)}-H^{(1)}\Delta g^{(1)} \tag{4-102}$$

由此可以设想 $\Delta H^{(1)}$的一种比较简单的形式为

$$\Delta H^{(1)}=\Delta x^{(1)}[q^{(1)}]^{t}-H^{(1)}\Delta g^{(1)}[w^{(1)}]^{t} \tag{4-103}$$

式中，$q^{(1)}$和 $w^{(1)}$为两个待定向量。将式（4-103）代入式（4-102）得

$$\Delta x^{(1)}[q^{(1)}]^{t}\Delta g^{(1)}-H^{(1)}\Delta g^{(1)}[w^{(1)}]^{t}\Delta g^{(1)}=\Delta x^{(1)}-H^{(1)}\Delta g^{(1)}$$

也就是说，应使

$$[q^{(1)}]^{t}\Delta g^{(1)}=[w^{(1)}]^{t}\Delta g^{(1)}=1 \tag{4-104}$$

考虑到 $\Delta H^{(1)}$应为对称阵，最简单的办法就是取

$$[q^{(1)}]=\eta_1\Delta x^{(1)} \tag{4-105}$$

$$w^{(1)}=\xi_1 H^{(1)}\Delta g^{(1)} \tag{4-106}$$

将式（4-105）和式（4-106）代入式（4-104）得

$$\eta_1[\Delta x^{(1)}]^{t}\Delta g^{(1)}=\xi_1[H^{(1)}\Delta g^{(1)}]^{t}\Delta g^{(1)}=1$$

若 $[\Delta x^{(1)}]^{t}\Delta g^{(1)}$以及 $[H^{(1)}\Delta g^{(1)}]^{t}\Delta g^{(1)}$不等于零，则有

$$\eta_1=1/\{[\Delta x^{(1)}]^{t}\Delta g^{(1)}\}=1/\{[\Delta g^{(1)}]^{t}\Delta x^{(1)}\}$$

$$\xi_1=1/\{[H^{(1)}\Delta g^{(1)}]^{t}\Delta g^{(1)}\}$$

于是得到

$$\begin{aligned}\Delta H^{(1)}&=\Delta x^{(1)}[q^{(1)}]^{t}-H^{(1)}\Delta g^{(1)}[w^{(1)}]^{t}\\&=\Delta x^{(1)}[\eta_1\Delta x^{(1)}]^{t}-H^{(1)}\Delta g^{(1)}[\xi_1 H^{(1)}\Delta g^{(1)}]^{t}\\&=\frac{\Delta x^{(1)}[\Delta x^{(1)}]^{t}}{[\Delta g^{(1)}]^{t}\Delta x^{(1)}}-\frac{H^{(1)}\Delta g^{(1)}[\Delta g^{(1)}]^{t}[H^{(1)}]^{t}}{[H^{(1)}\Delta g^{(1)}]^{t}\Delta g^{(1)}}\\&=\frac{\Delta x^{(1)}[\Delta x^{(1)}]^{t}}{[\Delta g^{(1)}]^{t}\Delta x^{(1)}}-\frac{H^{(1)}\Delta g^{(1)}[\Delta g^{(1)}]^{t}H^{(1)}}{[\Delta g^{(1)}]^{t}H^{(1)}\Delta g^{(1)}}\end{aligned}$$

从而可得

$$H^{(2)}=H^{(1)}+\Delta x^{(1)}[\Delta x^{(1)}]^{t}/\{[\Delta g^{(1)}]^{t}\Delta x^{(1)}\}-H^{(1)}\Delta g^{(1)}[\Delta g^{(1)}]^{t}H^{(1)}/\{[\Delta g^{(1)}]^{t}H^{(1)}\Delta g^{(1)}\}$$

一般有

$$H^{(k+1)}=H^{(k)}+\Delta x^{(k)}[\Delta x^{(k)}]^{t}/\{[\Delta g^{(k)}]^{t}\Delta x^{(k)}\}-H^{(k)}\Delta g^{(k)}[\Delta g^{(k)}]^{t}H^{(k)}/\{[\Delta g^{(k)}]^{t}H^{(k)}\Delta g^{(k)}\}$$

上述矩阵称为尺度矩阵，在整个迭代过程中它是在不断变化的。

2. 基本算法

① 选择初始近似点 $x^{(0)}$及判别收敛的正数 ε，令 $k\Leftarrow 0$。

② 若 $\|\nabla f(x^{(k)})\| \leqslant \varepsilon_1$ 或 $\|x^{(k+1)}-x^{(k)}\| \leqslant \varepsilon_2$，则停止迭代，$x^{(k)}$即为所求的解，否则进行③

③ 令 $H^{(1)} \Leftarrow I$，$k \Leftarrow 1$

④ 令 $s^{(k)}=-H^{(k)}\nabla f(x^{(k)})$

⑤ 进行一维搜索，即求 λ_k

$$f(x^{(k)}+\lambda_k s^{(k)})=\min_{\lambda} f(x^{(k)}+\lambda s^{(k)})$$

⑥ 令 $x^{(k+1)}=x^{(k)}+\lambda_k s^{(k)}$

⑦ 若 $\|\nabla f(x^{(k+1)})\| \leqslant \varepsilon_1$ 或 $\|x^{(k+1)}-x^{(k)}\| \leqslant \varepsilon_2$，则停止迭代，$x^{(k+1)}$即为所求的解，否则，当 $k=n$ 时，令 $x^{(1)}=x^{(n+1)}$，转③；当 $k<n$ 时，令

$$\Delta x^{(k)}=x^{(k+1)}-x^{(k)}$$

$$\Delta g^{(k)}=\nabla f(x^{(k+1)})-\nabla f(x^{(k)})$$

计算

$$H^{(k+1)}=H^{(k)}+\Delta x^{(k)}[\Delta x^{(k)}]^{t}/\{[\Delta g^{(k)}]^{t}\Delta x^{(k)}\}-H^{(k)}\Delta g^{(k)}[\Delta g^{(k)}]^{t}H^{(k)}/\{[\Delta g^{(k)}]^{t}H^{(k)}\Delta g^{(k)}\}$$

⑧ 令 $k \Leftarrow k+1$，转④。

（五）坐标轮换法

上述介绍的方法都需对目标函数进行求导运算，如当目标函数的解析表达式十分复杂或写不出其具体表达式的导数形式时，就无法使用上述方法。此时可使用本节及下一节将介绍的方法或以前介绍的单纯形法来进行寻优。

1. 基本思路

对于 n 维实数空间的目标函数 $f(x)$，本法的基本思路是依次沿其 n 个坐标轴方向进行一维寻优，经多次循环迭代后以达到最优解。

设 n 维坐标轴方向（坐标轴的单位向量）为

$$\begin{aligned}
e^{(1)}&=[1,0,0,\cdots,0]^{t}\\
e^{(2)}&=[0,1,0,\cdots,0]^{t}\\
e^{(3)}&=[0,0,1,\cdots,0]^{t}\\
&\vdots\\
e^{(n-1)}&=[0,0,\cdots,1,0]^{t}\\
e^{(n)}&=[0,0,\cdots,0,1]^{t}
\end{aligned}$$

为了简便，以后表示为

$$e^{(i)}=[0,0,\cdots,1,\cdots,0]^{t} \quad i=1,2,3,\cdots,n$$

2. 基本算法

① 选择初始近似点 $x^{(0)}$及判别收敛的正数 ε，令 $k \Leftarrow 1$。

② 沿方向 $e^{(k)}$进行一维搜索，得

$$f(x^{(k-1)}+\lambda_k e^{(k)})=\min_{\lambda} f(x^{(k-1)}+\lambda e^{(k)})$$

并令

$$x^{(k)}=x^{(k-1)}-\lambda_k e^{(k)}$$

③ 判别是否满足 $k=n$，若 $k=n$，则转④；若 $k<n$ 时，则令 $k \Leftarrow k+1$，转②。

④ 检验是否满足收敛性准则，即

$\|x^{(k)}-x^{(k-1)}\| \leqslant \varepsilon$，若满足判别标准，则停止迭代，$x^{(k)}$即为所求的解，否则令 $x^{(0)}=x^{(n)}$，

$k=1$，转②。

此法简单易懂，但收敛速度是不快的。

（六）Powell 法

1. 基本思路

Powell 法又称方向加速法，其基本思路是利用逐次构造共轭方向，并以此来加速收敛，故有方向加速法之称。Powell 法构造共轭方向勿需计算目标函数的导数，这是该法的一个显著性优点。

2. 基本算法

首先依坐标轮换法的思路，对 n 个互相正交的单位矢量 $e^{(i)}=[0,0,\cdots,1,\cdots,0]'(i=1,2,3,\cdots,n)$ 寻优，以得到最优点的一个初始近似点 $x^{(0)}$。令 $k=0$

① 令 $i=1$

② 计算 λ_i，使

$$f(x_{i-1}^{(k)}+\lambda_i s_i^{(k)})=\min_{\lambda} f(x_{i-1}^{(k)}+\lambda s_i^{(k)})$$

并令

$$x_i^{(k)}=x_{i-1}^{(k)}-\lambda_i s_i^{(k)}$$

③ 判别是否满足 $i=n$，若 $i=n$，则转④；若 $i<n$ 时，则令 $i \Leftarrow i+1$，转②

④ 求 λ_{i+1}使

$$f[x_0^{(k)}+\lambda_{n+1}(x_0^{(k)}-x_n^{(k)})]=\min_{\lambda} f[x_0^{(k)}+\lambda(x_0^{(k)}-x_n^{(k)})]$$

并令

$$x_{n-1}^{(k)}=x_0^{(k)}+\lambda_{n-1}(x_0^{(k)}-x_n^{(k)})$$

⑤ 计算

$$f(x_{m-1}^{(k)})-f(x_m^{(k)})=\max_{i=1,\cdots,n}[f(x_{i-1}^{(k)})-f(x_i^{(k)})]$$

以确定参数 m

⑥ 若有

$$\lambda_{n+1}<\{[f(x_0^{(k)})-f(x_{n+1}^{(k)})]/[f(x_{m-1}^{(k)})-f(x_m^{(k)})]\}^{1/2}$$

则令

$$s_i^{(k+1)}=s_i^{(k)} \qquad (i=1,2,\cdots,n)$$

转⑦，否则令

$$s_i^{(k+1)}=s_i^{(k)} \qquad (i=1,2,\cdots,m-1)$$

$$s_i^{(k+1)}=s_{i+1}^{(k)} \qquad (i=m,\cdots,n-1)$$

$$s_n^{(k+1)}=(x_0^{(k)}-x_n^{(k)})/\|x_0^{(k)}-x_n^{(k)}\|$$

转⑦。

⑦ 检验是否满足收敛性准则，即

$\|x^{(k)}-x^{(k-1)}\|\leqslant\varepsilon$，若满足判别标准，则停止迭代，$x^{(k)}$即为所求的解，否则

令

$$x_0^{(k-1)}=x_{n+1}^{(k)},$$

$$f(x_0^{(k+1)})=f(x_{n+1}^{(k)})$$

$$k=k+1$$

转①。

在此讨论的算法是所谓修改的 Powell 方法，实际上，修改的 Powell 方法也有多种，上面介绍的是 Sargent 修改方法。

三、全局优化算法

在化学研究的实践中，因所研究的对象千变万化，很多问题的目标函数非常复杂，而且还有很多问题根本就没有现存数学模型，本质上应属于全局最优问题，如简单采用上述方法时就很难奏效了，由于工程数学和计算机科学的飞速发展，近年来涌现了很多有效的全局最优新算法，这些算法的显著特点在于它们大都跳出了局部最优的框框，故很快在化学计量学中得到了广泛的应用，而且成为化学计量学近年来研究的一个热点。在此将对它们作专门的介绍。在这些新近涌现的全局优化方法中，对化学影响最大的应数模拟自然机理的模拟退火法[29~31]和遗传算法[32~35]，及由我国数学家提出的基于数论的优化方法[36,37]也将在此作出介绍。

（一）模拟退火法

1. 基本概念

1982 年，Kirkpatric 等将退火思想引入组合优化领域，提出一种解大规模组合优化问题的有效近似算法——模拟退火算法 (Simulated annealing algorithm)，它源于对固体退火过程的模拟，采用 Metropolis 接受准则，并用一组称为冷却进度表的参数控制算法进程，使算法在多项式时间里给出一个近似的全局最优解。可以这么说，固体退火过程的物理和统计性质是模拟退火算法的基础背景；Metropolis 接受准则使算法具有跳离局部最优的机制；而冷却进度表的合理选择是算法成功的重要前提。

固体在恒定温度下达到热平衡的退火过程，其统计性质可用 Monte Carlo 方法进行模拟，Monte Carlo 方法的特点是算法简单，但必须大量采样才能得到比较精确的结果，因而计算量很大。1953 年，Metropolis 等提出重要性采样法，他们用下述方法产生固体的状态序列：

先给定以粒子相对位置表征的初始状态 i，作为固体的当前状态，该状态的能量是 E_i，然后用摄动装置使随机选取的某个粒子的位移随机地产生一微小变化，得到一个新状态 j，新状态的能量是 E_j，如果接受准则 $E_j<E_i$，则该新状态就作为“重要”状态，如果 $E_j>E_i$，则考虑到热运动的影响，该新状态是否为“重要”状态，要依据固体处于该状态的几率来判断。由热力学的知识可知，固体处于状态 i 和 j 的几率的比值等于相应 Boltzmann 因子的比值，即

$$p=\exp[(E_j-E_i)/kT] \tag{4-107}$$

式中，k 为 Boltzmann 常数，T 为热力学温度。此几率为一个小于 1 的正实数。用随机数发生器产生一个 [0,1) 区间的随机数 ξ，若 $p>\xi$，则新状态 j 可作为重要状态，否则舍去。

若新状态 j 是重要状态，就以 j 取代 i 成为当前状态，否则仍以 i 为当前状态，再重复以上新状态的产生过程。在大量迁移后，系统趋于能量较低的平衡状态，固体状态的概率分布趋于 Gibbs 正则分布。

从以上讨论可知，高温下可接受与当前状态能差较大的新状态为重要状态，而在低温下只能接受与当前状态能差较小的新状态为重要状态，这与不同温度下热运动的影响完全一致。在绝对温度趋于零时，就不能接受任一 $E_j>E_i$ 的新状态了。

上述接受新状态的准则称为 Metropolis 准则，相应的算法称为 Metropolis 算法，这种算法的计算量将显著减少。

设一个优化问题的一个解 i 及其目标函数 $f(i)$ 分别与一个微观状态 i 及其能量 E_i 等价，

令随算法进程递减其值的控制参数 t 担当固体退火过程的绝对温度 T 的角色，则对于控制参数 t 的每一取值，算法持续进行“产生新解—判断—接受/舍弃”的迭代过程就对应着固体在某一恒定温度下趋于热平衡的过程，也就是执行了一次 Metropolis 算法。与 Metropolis 算法从某一初始状态出发，通过计算机的时间演化过程，求出系统最终达到的状态相似，模拟退火算法从某个初始解出发，经过大量解的变换后，可以求得给定控制参数值时优化问题的相对最优解，然后减小控制参数 t 的值，重复执行 Metropolis 算法，就可以在控制参数 t 趋于零时，最终求得优化问题的整体最优解。由于固体退火必须“徐徐”降温才能使固体在每一温度下都达到平衡状态，最终趋于能量最小的基态，控制参数的值也必须缓慢衰减，才能确保模拟退火算法最终趋于优化问题的整体最优解。

模拟退火算法用 Metropolis 算法产生优化问题解的序列，并由与 Metropolis 准则对应的转移概率 p

$$p_t(i \Rightarrow j)=\begin{cases}1 & f(i)<f(j)\\ \exp\{[f(i)-f(j)]/t\} & f(i)>f(j)\end{cases} \tag{4-108}$$

确定是否接受从当前解 i 到新解 j 的转移，式 (4-108) 中的 t 为控制参数。开始可让 t 取较大的值（与固体的熔解温度相对应），在进行足够多的转移后，缓慢减小 t 的值（与徐徐降温相对应），如此重复，直至满足某个停止准则时算法终止。因此，模拟退火算法可视为递减控制参数时 Metropolis 算法的迭代。可以证明，当寻优遵循此退火过程来进行，只要在过程的控制参数满足一定条件时，描述为有限非齐次马尔科夫链的模拟退火算法，将收敛于整体最优解[38,39]。

在介绍模拟退火算法之前，先介绍一下邻域结构与局部搜索的概念。

(1) 邻域结构　对于属于解空间 S 的每一个解 i，有一个解的集合 S_i，$S_i \subset S$，这些解在某种意义上是“邻近” i 的，集合 S_i 称为 i 的邻域，每个 $j \in S_i$ 称为 i 的一个邻近解。

(2) 局部搜索算法　局部搜索算法是从一个初始解 $i \in S$ 开始，然后运用一个解产生器，持续地在解 i（称为当前解）的邻域 S_i 中搜索比 i 更优的解，若找到比 i 更优的解，就用这个解取代 i，成为当前解，再对当前解的邻域进行搜索；否则，终止局部搜索。

2. 基本算法

设存在邻域结构和可能解产生器，再设 t_k 表示 Metropolis 算法第 k 次迭代时控制参数 t 的值，L_k 表示 Metropolis 算法第 k 次迭代时产生的变换个数，则模拟退火算法可由下列步骤组成：

① 初始化：设定初始解 i_0、初始控制参数 t_0 及初始变换个数 L_0

② $k=0$

③ 从 1 到 L_k，产生可能解集 $j_k(j=1,2,\cdots,L_k)$

④ 如果 $f(j_k)<f(i_k)$，则 $i_k \Leftarrow j_k$，否则计算：

$$p(j_k)=\exp\{-[f(j_k)-f(i_k)]/t_t\}$$

产生 [0,1) 的随机数 r

如果 $p(j_k)>r$，则记忆下 j_k 解，用于下一步搜索；

⑤ $k=k+1$

⑥ 确定下一步的 L_k，产生可能解集 $j_k(j=1,2,\cdots,L_k)$

⑦ 确定下一步的控制参数 t_k

⑧ 如果所得的当前解已满足某一终止标准，则停止计算，否则转③。

上述算法只是一个算法的基本步骤，其中每一步骤的具体确定与所需解决的具体问题密切相关，对于不同的问题，控制参数 t_k 和用于邻域搜索的可能解集的大小 L_k 都会不同。这一点是模拟退火算法不同于经典优化算法的重要标志，同时也给模拟退火算法留下了广阔发挥的天地。总的说来，一系列的控制参数 t_k 的确定对模拟退火算法是至关重要的，在模拟退火算法中，它有一个特殊的名称，称为冷却进度表。一般说来，较好的冷却进度表将得到较好的结果。这是因为，模拟退火算法的跳出局部最优的机制就完全是由 Metropolis 的按概率接受新状态的准则所提供的。

（二）遗传算法

1. 基本概念

遗传算法最早由 Holland[40] 于 20 世纪 60 年代创建，经过不断发展，近年来逐渐引起化学计量学家们的重视，被广泛用于解决化学上的难题。遗传算法是一种用计算机模拟生物自然进化过程搜索最优解的方法。它是受达尔文进化论的启发而产生的。自然界生物体通过自身的演化就能很好地适应环境，使问题得到完美的解决，这种才能让最好的计算机也相形见拙。大多数生物是通过自然选择和有性生殖这两种基本过程进行演化的。自然选择决定了群体中哪些个体能够存活并繁殖；有性生殖保证了后代基因中的混合和重组，比起单性生殖的简单基因拷贝产生的后代，有性生殖后代进化快得多。自然选择的原则是适应者生存，不适应者淘汰。具体地说，遗传算法是通过类比上述过程而产生的一种概率搜索算法，它是利用某种编码技术作用于称为染色体的数串，其基本思路是模拟由这些数串组成的群体的进化过程。遗传算法通过有组织地然而是随机的信息交换来重新结合那些适应性好的串，在每一代中，利用上一代串结构中适应性好的位和段来生成一个新的串的群体；此外，偶尔也要在串结构中尝试用新的位和段来替代原来的部分。遗传算法是一类随机算法，但它不是简单随机走动，它可以有效地利用已有信息来搜寻有希望改善解质量的串。类似于自然进化，遗传算法通过作用于染色体上的基因，寻找好的染色体来求解问题；与自然界相似，遗传算法对求解问题的本身一无所知，它所需要的仅是对算法产生的每个染色体进行评价，并基于适应值来选择染色体，使适应性好的染色体比适应性差的染色体有更多的繁殖机会。

遗传算法利用简单的编码技术和繁殖机制来表现复杂的现象，从而解决非常困难的问题。特别是由于它不受搜索空间的限制性假设的约束，不必要求诸如连续性、导数存在和单峰等假设，以及其固有的并行性，使它在化学中的应用前景正如 Lucasius 和 Kateman[41] 所说的，具有相当大的潜力。

遗传算法的过程可概括为 5 个基本步骤：① 按设定的规模产生第一代群体。群体中每条染色体的长度由所需优化的参数的个数决定。② 评价染色体的适应性。通过一定的方式将每条染色体中的参数值代入所求目标函数中，求出对应函数值。③ 将目标函数值经过转换，从而得出染色体的适应性优劣。④ 有选择地交叉染色体。每次随机地从母代染色体中选出两条染色体，随机地交换它们某些对应位置上的数段，从而产生下一代的新个体。在选母代染色体的过程中，每条染色体被选中的概率与其适应性成正比。⑤ 变异。按一定概率随机地对染色体上的基因进行变异，由此可以产生一些新的个体。整个算法在第 2 步和 5 步之间循环。

与前述的模拟退火算法类似，遗传算法的方法本身只能提供一个框架，在解决问题时，需密切依赖于它自身以外的过程，即必须与具体的问题结合在一起，经过精心设计，对应用到的特殊问题中的特定域的表示和专门知识加以充分利用。目前，通过一些化学计量学家的开拓性工作，遗传算法已开始被构造用于解决各种化学上的优化问题。Lucasius 和Kateman[41]用

遗传算法进行波长点的选择。在同时测定 4 种 RNA 中核苷酸时，由于它们的光谱相似，需对测定波长进行优化选择。文献［41］用光谱条件数作为优化目标函数，用 0、1 编码的染色体表示波长点的选择方案，在 36 个波长点中找出 6 个最佳点。Fontain[42]则将遗传算法用于寻找有机物同分异构体。将各个原子的标号编码为染色体，优化的目标函数为最小化学距离（minimum chemical distance），使化学距离值最小的结构即为最可能的同分异构体。计算结果与实际化学反应产物一致。另外，还有 Hibbert[43]用遗传算法计算化学动力学参数；Leardi 等人[44]用遗传算法进行特征选择；Brown[45,46]等人将此算法用于有机化合物分子结构的比较和检索；Lucasius[47]等用此算法进行聚类分析；李通化、丛培盛等人[48]则就数值遗传算法进行了有益探讨。以上方法在染色体的编码表示、目标函数的构造、交叉、变异的操作及限制条件等各个方面都大不相同，但所有的作者都力图找到最适合于解决该具体问题的方法。由于在化学领域引入遗传算法的时间尚短，目前仍有许多工作需要做。

2. 基本算法

如前所述，遗传算法通过模拟自然界生物“遗传→变异→适者生存”的进化过程，对优化空间进行随机搜索，从而得到全局最优解。数值遗传算法是将待优化的各个参数排列在一起，当作一条染色体，每个参数即为染色体中的遗传基因。根据染色体对环境的适应性（即其目标函数值的一个转换指标），通过各种遗传操作控制其繁殖情况，淘汰差的，保留好的，最终得到最好的染色体，亦即全局最优点。算法可用五个步骤描述如下。

（1）个体初始化　用待优化的指标组成染色体，如 $x_1x_2\cdots x_{n-1}x_n$，第一代个体由 20 条染色体组成，染色体中各参数值可在各自的约束区间内任取：

$$x_{ji}=\mathrm{limit}(i)\times r$$

式中，x_{ji}为第 j 条染色体中第 i 个参数，limit(i)为它的约束范围，r 为$[0,1)$之间均匀分布的随机数。

（2）求目标函数值　将每条染色体中的参数取值代入目标函数 $f(x)$中，求出对应函数值，若有两条或多条完全相同的染色体，保留一条不变，其余重新赋值。

（3）将目标函数转换为适应性　染色体优劣需通过适应性来评价。适应性由目标函数值求得。设为求极小化问题，故使目标函数值越小的染色体适应性越大，因此需进行函数值 f 与适应性 fittness 之间的转换。我们取目标函数的倒数 $f=1/\mathrm{fittness}$。

（4）选择性地染色体交叉　新一代的 20 条染色体通过对母代染色体进行“选择→交叉”而产生。在这个操作中，每次随机选择两条母代染色体，在它们中间随机地寻找一个切割点，交换其对应参数取值，得子代染色体。其过程如图 4-18。操作的总次数由参数 p_{mat}决定。每条染色体被选中的概率与其适应性成正比，这样好的基因就能有更大的机会通过交叉保留至下一代，本文决定概率的方法为

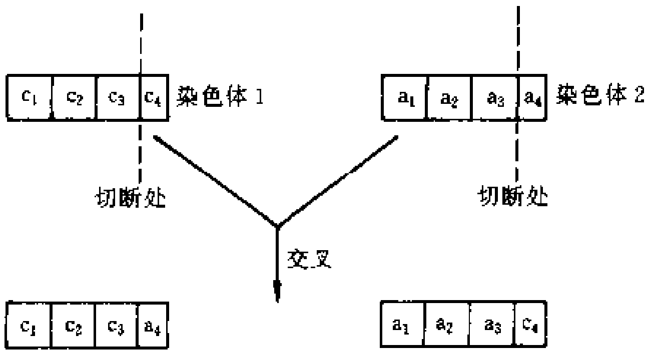


图 4-18　染色体交叉示意图

$$p(i)=\mathrm{fittness}(i)/\mathrm{sum}(\mathrm{fittness})$$

式中，$p(i)$为第 i 条染色体被选中的概率；fittness(i)为其适应性；sum(fittness)为整代染色体适应性之和。有时个别染色体会因适应性太大而过度繁殖，导致随后的搜索总在此染色体

决定的参数附近进行。为保持参数点在搜索空间的分散性，可定一个参数 β，当 $p(i)>\beta$ 时，令 $p(i)=\beta$。

(5) 变异　先按一定的概率 p_m，确定总共需要变异的基因数，然后随机地确定对哪些基因进行变异。变异方式有两种，一种是

$$\text{gene(new)}=\text{gene(old)}\pm r$$

即在原有基因上加减一个[0,1)之间的随机数。另一种是

$$\text{gene(new)}=\text{gene(old)}(1\pm r/100)$$

即在原有的点附近以较小步长进行搜索。

整个算法在第 (2) 步与第 (5) 步之间循环，直至目标函数收敛。

在遗传算法中，变异方式影响目标函数下降的速度，单用一种变异方式目标函数下降很慢，用两种方式可以明显地加速。两种方式选用的次数也很重要。可采取一种有针对性的方法：先用大的步长进行变异，当这种变异不再起作用时（即通过 Q 代繁殖，目标函数不下降），暂时保留这 20 条染色体，在后来的繁殖中选用第二种缩小了步长的变异方法，而当第二种方式不起作用时（即经过 W 代繁殖，目标函数不再下降），最后将分别将两种变异产生的 40 条染色体进行比较，从中选出好的 20 条，以后再用第一种变异，如此重复。在这里，第一种变异方式可以加快搜索速度，而第二种可以在参数附近进行更仔细的搜索。Q 和 W 均取 3 时，目标函数下降较快。

遗传算法参数的选择需根据问题而定，从原理上说，只要变异和交叉的概率大于 0，经过足够多的繁殖代数，算法最终总能收敛到全局最优，但参数的具体取值会影响收敛的速度。在一般的遗传算法问题中（如求最佳波长点，确定分子结构等），交叉的概率较大，一般 90%左右，变异的概率较小，仅 (1～2)%。但在数值遗传算法中，由于染色体交叉只能导致不同数值的重新组合，不能产生新基因，而我们需要对优化空间进行广泛的搜索，故需对基因进行强烈变异，经过比较发现，变异的概率 p_m 小于 0.5 时，算法收敛较慢。最终确定变异的概率 p_m 取第一种 0.5，第二种 0.8，交叉的概率 p_{mat} 取 0.5。

在遗传算法求极小化问题中，将目标函数转化成适应性这一步尚未受到重视。实际上选择适当的方法能够提高寻优速度，克服优化平台。我们模拟了一个函数

$$z=4/e^{(x/6)}+4/e^{(y/6)}+2\sin 2x+2\sin 2y+\cos(x/0.7)/(1+x)^{1.5} +\cos(y/0.7)/(1+y)^{1.5} \tag{4-109}$$

$$0<x<30,\quad 0<y<30$$

对式 (4-109) 采用两种不同的转化方法分别求极小值。一种为

$$\text{fittness}(i)=(\max(f)-f(i)+\min(f))/(\max(f)-\min(f)) \qquad i=1,2,\cdots,20 \tag{4-110}$$

$\max(f)$ 为从整代染色体中求出的最大目标函数值，$\min(f)$ 为最小目标函数值，$f(i)$ 为由第 i 条染色体求得的目标函数值。

另一种转化方法为

$$\text{fittness}(i)=1/f(i) \qquad i=1,2,\cdots,20 \tag{4-111}$$

终止准则为

$$|\min(f^{K+1})-\min(f^{K})|<0.00001$$

$\min(f^{K+1})$ 为从当代染色体中求出的最小目标函数值，$\min(f^{K})$ 为从上一代中求出的最小目

标函数值。

用每种方法运算 15 次，运算结果均与单纯型法结果一致。但对其染色体总繁殖代数进行假设检验，发现二者有显著性差别。说明计算达到同样精度，前者所需循环次数大于后者，亦即后者寻优速度快。其原因是由于该目标函数接近最优点时有优化平台，寻优达到此区域时，用式（4-110）的线性函数，不同染色体之间适应性差别将很小，因此差的染色体不能被淘汰；而式（4-111）中目标函数与适应性之间为非线性关系，当目标函数值越接近极小值时，适应性之间的细微差别越大，因此不同染色体之间的差别被放大，寻优将更集中地在好的染色体决定的参数点附近进行，优化速度加快。

（三）基于数论方法的序贯优化法

穷尽搜索是另一种类型的全局优化方法，该法遍历搜索空间中每一可能解，并将最好点作为全局最优解。理论上，穷尽搜索总能找到全局最优解。但是，它的严重缺陷是其优化效率十分低。然而，如果仅采用搜索空间中少量代表性解，而不是每一可能解，优化效率可望改善。数论方法（NTM），或称为伪蒙特卡洛法（Quasi MonteCarloMethods），是将数论和数值分析相结合的一种方法。数论方法最广泛的应用领域是数值积分、插值、积分等式、微分等式等的数值求解等问题。关于数论方法详细的综述可以参见 Korobov[49,50]、Niederreiter[51~53]和华罗庚、王元[54]的工作。数论方法的许多应用其实可以归结为一个核心问题，即如何找到数论网点。方开泰等人就是用数论方法（NTM）来均匀布点，提出了基于数论全局优化算法，并在此基础上，方开泰等[55]和 Niederreiter 等[56]提出了在序贯收缩的搜索空间内均匀布点的基于数论方法的序贯优化法（Sequential Number-Theoretic Optimization，SNTO）[57]。与其他全局优化算法相比，SNTO 具有原理清晰明了，算法易于实现等特点。在具体讨论 SNTO 以前，先介绍一下有关数论产生均匀布点的基本概念。

1. 数论网点（NT-net）

数论中将在一封闭有界的域 D 中均匀分布的点集称为 D 上的数论网点（NT-net）。事实上，数论网点可被定义为在 D 上服从均匀分布的代表点集。方开泰和王元的书中介绍了几种在单位立方体 C_s 上产生 NT-net 的方法。其中，通过好格子点（good lattice point set，glp 集）产生的 NT-net 易于计算。好格子点是通过 glp 产生表中的产生向量来构造的，具体过程如下：

① 从 glp 产生表中取出产生向量$(n;h_1,\cdots,h_s)$，n 为点集中的样本点数，s 为所考察空间的维数。

② 产生 glp 集$\{x_k,k=1,\cdots,n\}$，

$x_k=[\{(h_1k-0.5)/n\},\{(h_2k-0.5)/n\},\cdots,\{(h_sk-0.5)/n\}],k=1,\cdots,n$. 这里$\{\cdot\}$代表了取$\{\}$中的小数部分。方开泰和王元的论著[57]中给出了一系列 glp 产生表。以下仅列出二维和三维空间的 glp 产生表（表 4-45 与表 4-46）供读者参考。

表 4-45　二维 glp 产生表

n	8	13	21	34	55	89	144	233	377	610	987	…
h_1	1	1	1	1	1	1	1	1	1	1	1	…
h_2	5	8	13	21	34	55	89	144	233	377	610	…

表 4-46 三维 glp 表

n	35	101	135	185	266	418	597	828	1010	1220	1459	…
h_1	1	1	1	1	1	1	1	1	1	1	1	…
h_2	11	40	29	26	27	90	63	285	140	319	256	…
h_3	16	85	42	64	69	130	169	358	237	510	373	…

一般的域 D 上的 NT-net 可通过单位立方体 C_s 上的 NT-net 计算出来。例如，当 D 为 s 维实数空间中的长方体域 $[\boldsymbol{a}, \boldsymbol{b}]$ 时，设 $\{y_k, k=1,\cdots,n\}$ 是在单位立方体 C_s 上的 NT-net，对于给定的 $\boldsymbol{a}=(a_1,\cdots,a_s)$，$\boldsymbol{b}=(b_1,\cdots,b_s)$ 及 $\boldsymbol{y}_k=(y_{k1},\cdots,y_{ks})$，构造

$$x_{ki}=a_i+(b_i-a_i)y_{ki} \qquad i=1,\cdots,s \tag{4-112}$$

则 $\{\boldsymbol{x}_k=(x_{k1},\cdots,x_{ks}),k=1,\cdots,n\}$ 就是在 $[\boldsymbol{a},\boldsymbol{b}]$ 上的 NT-net。

2．基于 NT-net 的序贯优化算法（SNTO）

假设要搜寻满足下式的 $\boldsymbol{x}^*$

$$M=f(\boldsymbol{x}^*)=\min_{x\in D}f(\boldsymbol{x}) \tag{4-113}$$

式中，D 是 s 维实数空间中的封闭有界域，$f(\boldsymbol{x})$ 是定义在 D 上的连续函数。基于 NT-net 的优化思路如下：

取一在 D 上的 NT-net，$\bigcup\rho_n=\{\boldsymbol{x}_k,\ k=1,\ \cdots,\ n\}$。确定满足下式的 $\boldsymbol{x}_n^*$

$$M_n=f(\boldsymbol{x}_n^*)=\min_{1\leqslant k\leqslant n}f(\boldsymbol{x}_k) \tag{4-114}$$

式中，$\boldsymbol{x}_n^*$ 是 ρ_n 中使 $f(\boldsymbol{x})$ 达到最小值的点。由于 $f(\boldsymbol{x})$ 是连续函数且 D 封闭有界，容易证明当 $n\to\infty$ 时，$M_n\to M$。然而，理论和实践表明 M_n 到 M 的收敛效率不高。为了改善该法的效率，方开泰和王元提出了基于 NT-net 的序贯优化算法 SNTO[57]。

下面给出的是当 D 为 s 维实数空间中的长方体域 $[\boldsymbol{a},\boldsymbol{b}]$ 时的 SNTO 算法。

令 $\boldsymbol{a}=(a_1,\cdots,a_s)$，$\boldsymbol{b}=(b_1,\cdots,b_s)$，以及 $\boldsymbol{x}=(x_1,\cdots,x_s)$，SNTO 的过程如下：

① 初始化。令 $t=0$，$D^{(0)}=D$，$\boldsymbol{a}^{(0)}=\boldsymbol{a}$ 以及 $\boldsymbol{b}^{(0)}=\boldsymbol{b}$。

② 产生 NT-net。用 glp 集产生在 $D^{(t)}=[\boldsymbol{a}^{(t)},\boldsymbol{b}^{(t)}]$ 上均匀分布含 n_t 点的点集 $\rho^{(t)}$。

③ 计算近似最优点。确定满足下式的 $\boldsymbol{x}^{(t)}\in\rho^{(t)}\cup\{\boldsymbol{x}^{(t-1)}\}$ 和 $M^{(t)}$：

$$M^{(t)}=f(\boldsymbol{x}^{(t)})\leqslant f(\boldsymbol{y}),\forall\ \boldsymbol{y}\in\rho^{(t)}\cup\{\boldsymbol{x}^{(t-1)}\} \tag{4-115}$$

式中，$\boldsymbol{x}^{(-1)}$ 是空集，$\boldsymbol{x}^{(t)}$ 和 $M^{(t)}$ 是到目前为止 $\boldsymbol{x}^*$ 和 $\boldsymbol{M}$ 的最佳近似。

④ 终止准则。令 $\boldsymbol{c}^{(t)}=(\boldsymbol{b}^{(t)}-\boldsymbol{a}^{(t)})/2$。如果 $\max\boldsymbol{c}^{(t)}<\delta$，$\delta$ 为一预置的较小数，则可认为 $D^{(t)}$ 收缩到足够小，终止算法并接受 $\boldsymbol{x}^{(t)}$ 和 $M^{(t)}$。否则，执行下一步。

⑤ 收缩搜索域。构造新的搜索域 $D^{(t+1)}=[\boldsymbol{a}^{(t-1)},\boldsymbol{b}^{(t-1)}]$：

$$\boldsymbol{a}_i^{(t+1)}=\max(\boldsymbol{x}_i^{(t)}-\gamma\boldsymbol{c}_i^{(t)},\boldsymbol{a}_i) \tag{4-116}$$

且

$$\boldsymbol{b}_i^{(t+1)}=\min(\boldsymbol{x}_i^{(t)}+\gamma\boldsymbol{c}_i^{(t)},\boldsymbol{b}_i) \tag{4-117}$$

式中，γ 是一预置的收缩因子。令 $t=t+1$，转到②。

方开泰等建议选取 $n_1>n_2=n_3=\cdots$ 并使用 $\gamma=0.5$。此外，虽然以上算法是为长方体搜索域中的优化问题而设计的，但可很容易地将它推广至一般的搜索域中。图 4-19 大意地示出了 SNTO 法的优化过程。图 4-19 中 $D^{(t)}$ 的每一边长是 $D^{(t-1)}$ 的每一边长的 1/2。然而，如果 $\boldsymbol{x}^{(t)}$ 是 D 上的边界点，则要求 $D^{(t)}$ 落在 D 内。因此，边长有可能收缩到小于 $1/2D^{(t-1)}$ 中边长。但

$D^{(t+1)} \subset D^{(t)} (t>1)$并不一定总是成立，如 $D^{(3)} \subset D$ 但 $D^{(3)} \not\subset D^{(2)}$。

由于 SNTO 也如模拟退火和遗传算法一样，具有跳出局部最优的能力，因此它也能搜索到全局最优。有关 SNTO 法全局最优能力的论述可参考 Niederreiter 等[51]和方开泰等人[58]的工作。

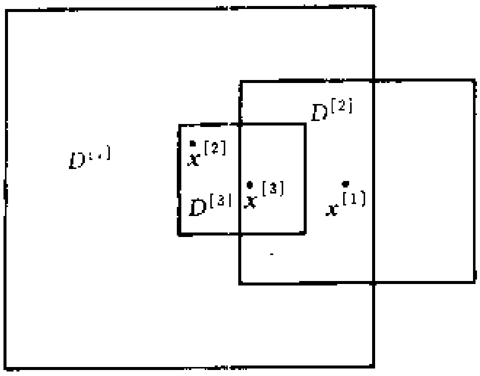


图 4-19　SNTO 序贯搜索示意图

搜索空间中的第一次布点数对 SNTO 法的全局优化效果来说至关重要。推荐采用 $n_1 > n_2 = n_3 = \cdots$，是十分合理的，这是由于起始搜索空间最大，因此，起始搜索空间中的布点数应较多。n_1 决定了一最好点，第二大搜索区域正落在该点附近。可见，需仔细选择第一次的布点数 n_1。通常来说，n_1 越大，算法收敛到全局最优解的可能性就越大，但对于一个给定的优化问题，n_1 的值无法事先给出，所需 n_1 的大小应由特定问题目标函数响应曲面的复杂程度来决定。例如，响应曲面越陡，所需 n_1 就越大。因此，n_1 的选择应该能以最少的布点探测尽可能多的全局最优可能域（可近似认为是单峰区间）。我们推荐用逐步增加的 n_1 来进行试验。如果增加布点数所得到的最好点和以前得到的最好点落在同一单峰区间，则可近似认为进一步增加布点数对优化结果改善不大。优化中的一个经验规则是当两点处于同一单峰区间时，它们具有相近的函数值且距离较近。因此，这两个指标可用以衡量第一次布点中找到的最好点的优度。为此，提出了如下的选择 n_1 的方法：

令 $n_1(1) < n_1(2) < \cdots$，其中所有的 $n_1(i)$ 均取自 glp 集产生表中。

① 令 $i=1$。

② 产生在原始搜索域 D 上均匀分布的含 $n_1(i)$ 点的点集 $\rho^{(i)}$，确定满足下式的 $\boldsymbol{x}^{(i)} \in \rho^{(i)}$ 和 $M^{(i)}$：

$$M^{(i)} = f(\boldsymbol{x}^{(i)}) \leqslant f(\boldsymbol{y}), \forall\ \boldsymbol{y} \in \rho^{(i)} \tag{4-118}$$

③ 如果 $i=1$，令 $i=i+1$ 并转到②；否则，计算 $V_{i-1} = |M^{(i)} - M^{(i-1)}|$ 和 $S_{i-1} = \| \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i-1)} \|$。

④ 如果 V_{i-1} 和 S_{i-1} 都降至较小，则可选取 $n_1(i)$ 或 $n_1(i-1)$ 作为 n_1；否则，令 $i=i+1$，转到②。

笔者用两个含有多个局部最优点和一个全局最优点的复杂数学函数，即

$$1x^2 + 2y^2 - 0.3\cos(3\pi x) - 0.4\cos(4\pi y) + 0.7 \qquad -1 \leqslant x, y \leqslant 1 \tag{4-119a}$$

和

$$1 + \sin^2(x) + \sin^2(y) - 0.1\exp(-x^2 - y^2) \qquad -10 \leqslant x, y \leqslant 10 \tag{4-119b}$$

对 SNTO 算法进行了研究。由式 (4-119) 表出的函数具有位于原点的全局最优和一些局部最优，有关此函数的三维图和等高线图示于图 4-20。对于该函数，用变步长通用模拟退火法从两个不同的起始点出发各运行了 4 次（表 4-47），与 Kalivas 等人所得结果一致[59]，VSGSA 能收敛到函数 GSA1 的全局最优解。在用 SNTO 优化时，每次优化过程中 N 次布点中的布点数 $n_i (i=1,\cdots,N)$ 均相同，对不同的 n_i 测试了 SNTO 的优化效果（表 4-47）。从表 4-47 可见，除了对应于 $n_i = 21 (i=1,2,\cdots,N)$ 的优化过程以外，其余各次优化均能收敛到全局最优解。正如前面部分所指出的，n_1 的选择影响 SNTO 的优化效果，因此选取 n_1 以保证算法的收敛结果具有较好的可信度十分关键。表 4-48 列出了通过测试试验选择 n_1 的结果。$V_5 (= |M^{(i)} - M^{(i-1)}|)$

和 $S_5(=\|x^{(t)}-x^{(t-1)}\|)$ 均降至较小，从而暗示第一次布点数为 $n_1=55$ 或 $n_1=89$ 时、得到的最好点很可能落在同一单峰区间。因此可以选取 $n_1=55$ 或 $n_1=89$。虽然 $n_i=8$ 的优化过程也能以很少的布点数收敛到全局最优解，但这是由于所布的点恰好落在全局最优解附近所致，所得的结果并不可靠。

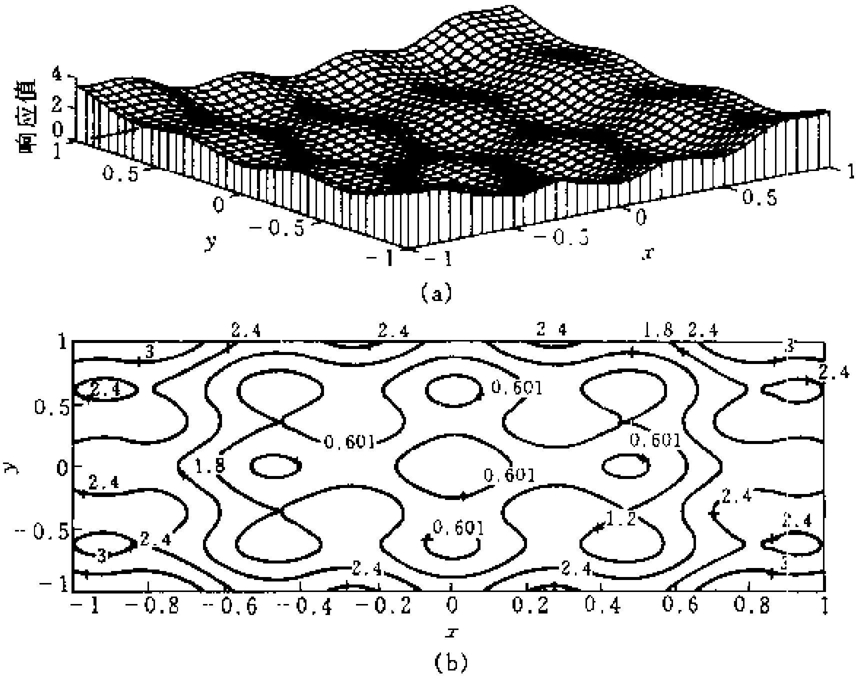


图 4-20　由式（4-119a）表示的函数的三维图（a）和等高线图（b）

表 4-47　用 VSGSA and SNTO 对由式（4-119a）表示的函数所得优化结果①

（全局最优坐标点为(0,0)；最优相应为 0）

VSGSA 结果①

初始 β	起始搜索点	运行次数	所得最优坐标点	最优响应	目标函数运算次数
0.80	(0.85,0.85)	1	$(5.0\times10^{-5},-2.0\times10^{-5})$	5.0×10^{-8}	835
		2	$(4.4\times10^{-5},-2.4\times10^{-5})$	4.7×10^{-8}	1450
		3	$(-5.4\times10^{-5},-1.4\times10^{-5})$	4.8×10^{-8}	815
		4	$(2.9\times10^{-5},-2.9\times10^{-5})$	4.1×10^{-8}	1533
1	(−0.50,−0.50)	1	$(5.3\times10^{-5},-2.2\times10^{-5})$	5.7×10^{-8}	815
		2	$(-5.4\times10^{-5},1.4\times10^{-5})$	4.8×10^{-8}	959
		3	$(5.1\times10^{-5},2.2\times10^{-5})$	5.4×10^{-6}	1081
		4	$(1.1\times10^{-5},5.1\times10^{-5})$	9.0×10^{-9}	915

SNTO 结果②

$n_i(i=1,\cdots,V)$	总迭代数	所得最优坐标点	最优响应	目标函数运算次数
8	15	$(7.6\times10^{-5},7.6\times10^{-6})$	2.8×10^{-9}	120
21	14	(0.0714,0)	0.0706	294
55	15	$(0,-2.7\times10^{-15})$	0	825
89	15	$(6.7\times10^{-17},8.0\times10^{-16})$	0	1335

① 初始步长为 0.15，步长以 2 为因子来减少或增加，初始 β 同文献[59]，当步长小于 1.0×10^{-4} 时，VSGSA 终止运算。

② 当搜索区间长度小于 1.0×10^{-4} 时，SNTO 终止运算。

表 4-48　对由式（4-119a）表示的函数进行选择 n_1 的试验结果（一）①

i	$n_1(i)$	响应值	V_{i-1}	S_{i-1}
1	8	0.6321		
2	21	0.4395	0.1926	0.7075
3	55	0.2394	0.2001	0.5040
4	89	0.2464	0.007	0.0799

① $V_{i-1}=|M^{(i)}-M^{(i-1)}|$，$S_{i-1}=\|x^{(i)}-x^{(i-1)}\|$；$x^{(i)}$和$M^{(i)}$分别为所得最优相应和最优相应所对应的坐标点。

由式(4-119b)表示出的函数具有许多局部最优和一位于原点的全局最优，其函数的响应表面远比前一个函数的响应表面复杂，此函数的三维图和等高线图示于图 4-21。对于该函数，VSGSA 从 4 个不同的起始点出发各运行了 2 次，表明 VSGSA 的结果不够满意，在所有的 8 次优化过程中，只有两次找到了全局最优解。Kalivas 等人的研究显示，对该函数进行优化时通用模拟退火法（GSA）也易于陷入局部最优。表 4-49 列出了 SNTO 对该函数优化时，试验选择 n_1 的情况。由于该函数全局最优响应值和局部最优响应值的差别不十分明显，因而 V_i 间的差别不显著，主要是利用 S_i 来选取 n_1。表 4-49 表明在第一次布点中 $n_1=610$ 个点中的最好点和 $n_1=987$ 个点中的最好点间的距离 $S_5=0.3010$，该数值要比 $S_4=4.4033$ 小得多。因此，n_1 可选为 610 或 987。同时，表 4-50 中 SNTO 的优化结果表明，不适当的 n_1 会导致 SNTO 收敛到局部最优，例如，当 $n_1=89$ 或 $n_1=233$ 时。与此相反，所有 $n_1=610$ 的优化过程均收敛

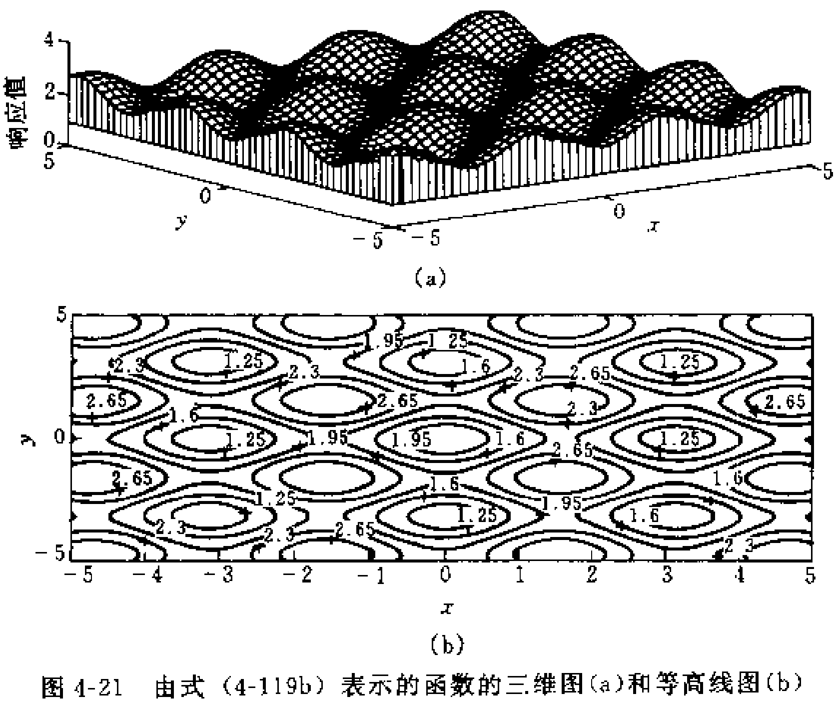


图 4-21　由式（4-119b）表示的函数的三维图(a)和等高线图(b)

表 4-49　对由式（4-119a）表示的函数进行选择 n_1 的试验结果（二）①

i	$n_1(i)$	最好点的响应	V_{i-1}	S_{i-1}
1	89	1.0001		
2	144	0.9106	0.0895	6.2230
3	233	1.0108	0.1002	8.7029
4	377	1.0003	0.0105	4.3749
5	610	0.9006	0.0997	4.4033
6	987	0.9980	0.0974	0.3010

① 所用符号意义同表 4-48。

表 4-50 用 VSGSA and SNTO 对由式（4-119b）表示的函数所得优化结果①

（全局最优坐标点为（0，0）；最优相应为 0.9）

VSGSA 结果①

初始 β	起始搜索点	运行次数	最终步长	所得最优点坐标	所得最优相应	目标函数运算次数
0.4	(−5.62, −9.06)	1	9.2×10^{-5}	(−6.283, 6.283)	1.000	3416
		2*	4.7×10^{-2}	(−3.186, 3.144)	1.002	5792
−0.4	(1.32, 5.76)	1	9.2×10^{-5}	$(7.1\times10^{-5}, 4.7\times10^{-5})$	0.900	657
		2	9.2×10^{-5}	(6.2831, 0)	1.000	8523
0.35	(4.05, −3.27)	1	9.2×10^{-5}	$(-6.4\times10^{-5}, -3.1\times10^{-5})$	0.900	2348
		2*	3.7×10^{-1}	(9.425, 3.141)	1.000	2977
0.35	(−6.31, 1.60)	1*	5.9×10^{-3}	(3.144, 6.285)	1.000	8297
		2*	4.7×10^{-2}	$(-5.2\times10^{-2}, 9.320)$	1.012	6270

SNTO 结果②

n_1	$n_i(i=2,\cdots,N)$	所得最优相应	所得最优点坐标	目标函数运算次数
89	89	1.000	(−6.2832, 0)	1869
144	144	0.900	$(6.6\times10^{-8}, 6.6\times10^{-8})$	3024
233	233	1.000	(−6.2832, −6.2832)	4893
377	377	0.900	$(-9.3\times10^{-8}, 9.3\times10^{-8})$	7917
610	144	0.900	$(1.8\times10^{-7}, 1.8\times10^{-7})$	3490
610	89	0.900	$(-9.6\times10^{-7}, 3.3\times10^{-7})$	2390
610	8	0.900	$(3.3\times10^{-6}, 3.3\times10^{-6})$	770

① 运行条件同表 4-47。

② 所有迭代次数均为 21；其余运行条件同表 4-47。

到全局最优解。在一个极端的情况下，$n_1=610$ 和 $n_1=8(i=2,\cdots,21)$的优化过程也找到了全局最优解。可见，在用 SNTO 对复杂函数优化的过程中，选取 $n_1>n_2=n_3\cdots$是十分有效的。较大的 n_1 有助于使起始邻域落在全局最优附近，而相对较小的 $n_i(i>1)$则有助于减少总计算量。需要指出的是，和前面的例子类似，虽然 $n_i=144(i=1,2,\cdots,21)$的优化过程也找到了全局最优，但其结果并不可信。

将此法用于灰色分析体系定量分析（参见第六章），在存在未知干扰下直接测定待测组分的浓度，得到了较好的结果。结果表明 SNTO 和 VSGSA 对混合物的定量结果无显著差异。但 VSGSA 评估目标函数值的次数明显多于 SNTO[60]。

参 考 文 献

1 Morgen E D. Chemometrics: Experimental Design, John Wiley & Sons, 1991

2 Box G E P, Hunterand W G, Hunter J S. Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building. Wiley, 1978

3 Bayneand C K, Rubin I B. Practical Experimental Designsand Optimization Methods for Chemists, VCH, 1986

4 Deming S N, Morgan S L. Experimental Design, A Chemometrics Approach, Elservier, 1987

5 Boxand G E P, Hunter J S. Technometrics, 1961; 3: 311～429

6 Bisgaard S. Technometrics, 1994; 36: 308～311

7 Srivastava J N. Mordern Factorial Design for Experimenters and Statisticians, in Statistical Designand Analysis of Industrial Experiments, Edited by Subir Ghosh, Marcel Dekker inc., New York and Basel, 1990: 311～405

8 项可风，吴启光. 试验设计与数据分析. 上海：上海科学技术出版社，1989

9 方开泰. 均匀设计与均匀设计表. 北京：科学出版社，1994

10 王元，方开泰. 科学通报，1981；26：485～489

11 Bundschuhand P. Zhu Y C. A method for exact calculation of the discrepancy of two-dimensional finite pointsets (I). Abhandlungenausdem Math., Seminarder Univ. Hamberg, Bd. 63
12 Fang K T and Li J K. Technique Report. Math-042. Hong Kong: Hong Kong Baptist College
13 丁元.应用概率统计,1986;2:153~160
14 蒋声,陈瑞琛. 高校应用数学学报,1987:(4):2
15 陈瑞琛. 高校应用数学学报,1989;(4):4
16 方开泰,郑胡灵. 应用概率统计,1992;8:2~16
17 朱伟勇,段晓东,唐明等. 最优设计在工业中的应用. 沈阳:辽宁科学技术出版社,1993
18 Fedorov V V. Theory of Optimal Experiments, Acaolemic, 1976
19 Antikon A C and Donev A N. Optimum experimental Designs. Clarendon Press, Oxford, 1992
20 朱伟勇等. 最优设计理论与应用. 沈阳:辽宁人民出版社,1981
21 Ernst S N. Rev Sci Instrum, 1968; 39: 998
22 Deming S N, Parker Jr L R. Crit Rev in Anal Chem,1978; 7: 187
23 Nelderand J A, Mead R. Comput J, 1965; 7: 308
24 关颖男编著. 混料试验设计. 上海:上海科技出版社,1990
25 王元,方开泰. 中国科学,1997
26 赵凤治. 数值优化中的二次逼近法. 北京:科学出版社,1994
27 《运筹学》试用教材编写组. 运筹学. 北京:清华大学出版社,1982
28 余俊,廖道训. 最优化方法及其应用. 武汉:华中工学院出版社,1984
29 康立山,谢云,尤矢勇等. 非数值并行算法. 第一册. 模拟退火算法. 北京:科学出版社,1994
30 Aarts E H L, Korst J H M. Simulated annealing and Boltzmann machines. John Wiley and Sons, 1989
31 Kalivas J H. Adaption of Simulated Annealing to Chemical Optimization Problems: Elsevier, Amsterdam. 1995
32 刘勇,康立山,陈毓屏. 非数值并行算法(第一册). 遗传算法. 北京:科学出版社,1995
33 Goldberg D E. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA. 1989
34 Lucasius C B, Kateman G. Trends in Anal Chem, 1991; 10: 254
35 Hilbert D B. Chemom Intell Lab Sys, 1993;19:319
36 Fang K T and Wang Y. Number-theoretic Methods in Statistics. Chapman & Hall London, 1993
37 陈文灿,崔卉,梁逸曾等. 化学学报,1997;55:693~702
38 Yao Xin, Li Guojie. General simulated annealing, J of Comput Sci & Technol, 1991; 6: 329~338
39 Van Laarhoven P J M, Laarts E H. Simulated annealing: Theory and applications, D Reidel, 1987
40 Hibbert D B. Chemom Intell Lab Syst. 1993; 19: 277
41 Lucasius C B, Kateman G. Trends in Analytical Chemistry, 1991; 10: 254
42 Fontain E. J Chem Inf Comput Sci, 1992; 32: 748
43 Hibbert D B. Chemom Intell Lab Syst, 1993; 19: 319
44 Leardi R, Boggia R, Terrile M J. Chemometrics, 1992; 6: 267
45 Brown R D, Jones G, Willett P et al. J Chem Inf Comput Sci, 1994; 34: 63
46 Brown R D, Jones G, Willett P et al. J Chem Inf Comput Sci, 1994; 34: 47
47 Lucasius C B, Dane A D, Kateman G. Anal Chim Acta. 1993; 282: 231
48 Cong P, Li T. Anal Chim Acta, 1994; 293: 191
49 Korobov N M. Dokl Akad Nauk SSSR, 1959; 124: 1207~1210
50 Korobov N M. Trigonometric Sums and Their Applications, Nauka, Moscow. 1989
51 Niederreiter H. Bull Amer Math Soc, 1978; 84: 957~1041
52 Niederreiter H. Quasi-Monte Carlo Methods for Multi-dimensional Numerical Integration, International Series of Numerical Mathematics, Birk. Springer-Verlag, 1988
53 Niederreiter H. Random Number Generation and Quasi-Monte Carlo Methods, SIAM, Philadelphia, 1992
54 Hua L K, Wang Y. Application of Number Theory to Numerical Analysis, Berlin: Springer. 1981

55 Fang K T, Wang Y. in Lecture Notes in Contemporary Mathematics (Yang L and Wang Y eds), Beijing: Science Press, 1990: 17～28

56 Niederreiter H, Peart P. SIAM J Sci Statist Comput, 1986; 7: 660～664

57 Fang K T, Wang Y. Number-theoretic Methods in Statistics. London: Chapman & Hall, 1993

58 Fang K T, Wang Y and Bentler P M. Statistical Science, 1994; 9: 416～428

59 Sutter J M, Kalivas J H. Anal Chem, 1991; 63: 2386～2390

60 Zhang L, Liang Y Z, Yu R Q and Fang K T. J Chemometrics, 1997; 11: 267～281

第五章　分析检测理论与信号处理方法

分析检测理论与信号处理方法主要包括两个方面的内容：① 有关分析信号的量测误差、检测下限和统计检验，这一部分主要来自数理统计的理论和方法，属于经典的统计学范畴[1,2]；② 分析仪器信号的预处理，这一部分涉及现代信号处理的理论和方法和近年来发展的小波分析新方法，均属于本章讨论的内容[3,4]。

本章将先对分析方法的检测限、分析数据的统计检验和统计推断和方差分析作出较详细的介绍。包括涉及到的经典的统计学及相关计算机程序。在分析信号的预处理方法的介绍方面，对于一般常用的经典方法，如分析信号的平滑方法、分析信号的求导方法、分析信号的变换方法都将给出必要的介绍，但主要着重点将放在介绍目前已经或开始受到化学计量学研究重视的多变量处理的新方法之上，即基于卡尔曼滤波和小波变换的信号预处理方法．在详细介绍这些方法的基本原理的基础上，对它们的基本思路和基本算法亦将给出详细介绍，并给出相应的计算机程序和计算时所需的数据表格。

第一节　分析方法的检测限

定性分析的任务是检测某种成分在试样中是否存在。定性分析中能检测的最低含量，以“检测限”表征，有关分析方法检测限的定义，曾是分析化学中长期争议的问题。国际纯粹与应用化学联合会（International Union of Pure and Applied Chemistry，IUPAC）建议的检测限定义[5]认为，以浓度 c_L 或物理量 q_L 表述的检测限，是由用给定的分析方法能以一定的准确度检出的最小测度 x_L 导出的。这一概念后被简化为“检测限是给定分析方法能可靠地检出的最低浓度”（美国化学会）[6]。如何度量上述定义中的“一定的确定度”及“可靠地检出”这些概念？这只能建立在统计理论基础之上，因此，分析检验理论是定性检定的统计理论基础。

在统计检测理论基础上，可给出分析方法检测限的严谨定义。设有分析信号 y，由其真值 η 与噪声或误差 δ 组成

$$y=\eta+\delta$$

δ 服从均值为零，标准差为 σ 的正态分布，即 $\delta\sim N(0,\sigma)$。要检测这一分析信号，需考察空白信号与分析信号两者的概率密度函数。y 的产生有两个来源：待测组分的存在（其浓度 $c>0$），分析仪器的噪声（仪器噪声）和/或分析反应的空白值（化学噪声），即 $c=0$ 时的空白信号 y_0。以最简单的线性模型为例，y 和 c 的关系可写成

$$y=y_0+bc$$

b 为线性校正模型中的斜率或灵敏度。

从统计上讲，模型涉及两个概率密度函数，空白信号的概率密度函数 $P_0(y)$是与 $c=0$ 对应的；$P_1(y)$则是对应于分析信号的概率密度函数（$c>0$，见图 5-1），$y-y_0$ 为净分析信号。

确定分析检测限等价于选择下述两种假设判决之一：

H_0——测得的信号属于分布 $P_0(y)$，即信号来源于空白噪声，判决待测的组分不存在。

H_1——测得的信号属于分布 $P_1(y)$，即信号来源于待测组分，判决待测的组分存在。

进行 H_0 和 H_1 的选择时，需有一定的统计检测准则，以确定信号指标轴上某一称为判决

水平或检测水平 y_k 的值，作为将信号域划分为 R_0 和 R_1 的分界（见图 5-1）。$y \leqslant y_k$ 时（R_0 域）选择 H_0，$y \geqslant y_k$ 时（R_1 域）选择 H_1，这种选择正确与否，可用下述概率描述：

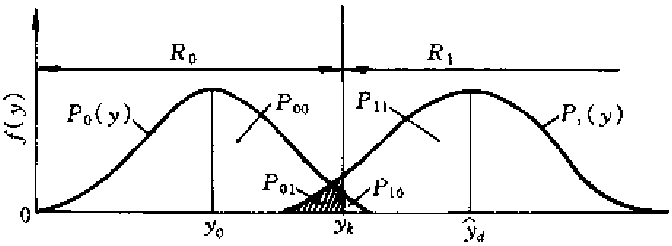


图 5-1　空白与分析信号的概率密度函数

（1）根据分析信号选择 H_0，且待测组分确实不存在，其概率 P_{00} 为

$$P_{00} = \int_{-\infty}^{y_k} P_0(y)\mathrm{d}y = \int_{R_0} P_0(y|H_0)\mathrm{d}y$$

式中，$P_0(y|H_0)$ 表示的是 H_0 是正确的条件下信号的条件概率分布，亦记为 $P_0(y)$，下标 0 表示信号来源于空白噪声。

（2）待测组分确实不存在，误选择 H_1（第一类错误，即以假为真），其概率 P_{10} 为

$$P_{10} = \int_{y_k}^{\infty} P_0(y)\mathrm{d}y = \int_{R_1} P_0(y|H_0)\mathrm{d}y = 1 - P_{00}$$

（3）待测组分确实存在，而误选择 H_0（第二类错误，即以真为假），其概率 P_{01} 为

$$P_{01} = \int_{-\infty}^{y_k} P_i(y)\mathrm{d}y = \int_{R_0} P_i(y|H_1)\mathrm{d}y$$

在 H_1 是正确的条件下，信号的条件概率记为 $P_i(y)$，下标 i 表示存在待测组分。

（4）待测组分确实存在，选择 H_1，其概率 P_{11} 为

$$P_{10} = \int_{y_k}^{\infty} P_i(y)\mathrm{d}y = \int_{R_1} P_i(y|H_1)\mathrm{d}y = 1 - P_{01}$$

在信号检测理论中，提出了许多检测准则，其中较有代表性的是 Bayes 准则。Bayes 准则要求知道 H_0 和 H_1 的验前概率 q 和 $1-q$，并需给出上述 4 种决策下的风险 C_{00}、C_{10}、C_{01}、C_{11}，这样就可以计算平均风险 C

$$C = q\left[C_{00}\int_{y_k}^{\infty} P_0(y)\mathrm{d}y + C_{10}\int_{y_k}^{\infty} P_0(y)\mathrm{d}y\right]$$
$$+ (1-q)\left[C_{01}\int_{-\infty}^{y_k} P_i(y)\mathrm{d}y + C_{11}\int_{-\infty}^{y_k} P_i(y)\mathrm{d}y\right]$$

使得上式的平均风险最小的判决界即可视为 Bayes 判决检测限。

在分析检验中较适宜采用的判决准则是 Neyman-Pearson 准则[7]。在分析检测中，例如产品质量检验，如根据有某种杂质存在于产品中作出“产品不合格”的判决，则生产必须停止，这时如系误判，将造成一定损失，应限制 P_{10} 在一定范围之内。Neyman-Pearson 准则的基本思想即是，事先赋予 P_{10} 某一给定数值以确定判决水平 y_k。

如设分析信号与空白噪声均服从正态分布，则有

$$P_{10} = \int_{y_k}^{\infty} P_0(y)\mathrm{d}y = (1/\sqrt{2\pi}\sigma)\int_{y_k}^{\infty}\exp[-(y-y_0)^2/(2\sigma^2)]\mathrm{d}y$$

在此，y_0 是空白信号（噪声）的均值，σ 为其标准差。作变换，令 $z=(y-y_0)/\sigma$，并记 $(y_k-y_0)/\sigma$ 为 z_k，得

$$P_{10} = (1/\sqrt{2\pi})\int_{y_k}^{\infty} \exp(-z^2/2)\mathrm{d}y$$

则判决水平(或称检测水平)y_k 为

$$y_k = y_0 + z_k\sigma$$

即当 $y \leqslant y_k$ 时(R_0 域)判决 H_0,否则判决 H_1。在此对于分析信号 y,都假设其标准差与空白信号的标准差相同,即均为 σ。如平行测定 n 次,得分析信号的均值 $y_{k(n)}$,其标准差将为 $\sigma_y = \sigma/\sqrt{n}$。

$$y_{k(n)} \leqslant y_0 + z_k\sigma/\sqrt{n} \qquad (\text{判决 } H_0)$$

$$y_{k(n)} \geqslant y_0 + z_k\sigma/\sqrt{n} \qquad (\text{判决 } H_1)$$

z_k 值可由给定的 P_{10} 值借 Laplace 函数或从标准正态分布表中求出。例如,P_{10} 取 0.01,即第一类错误概率为 1%时

$$P_{10} = 0.01 = (1/\sqrt{2\pi})\int_{z_k}^{\infty} \exp(-z^2/2)\mathrm{d}z = 1 - F(z)$$

由标准正态分布表可查出 $z_k = 2.32$,已知空白信号均值 y_0 及其标准差 σ,即可确定判决水平,在给定的 P_{10} 值的条件下作出待检测物质是否存在的判决。

在统计检测理论基础上,可给出分析方法检测限的严谨定义。现试将分析信号 y 与噪声 y_0 的概率分布图与对应的待测物浓度相比较加以考察(图 5-2)。首先按 Neyman-Pearson 准则给定 P_{10} 值以找出 $y = y_k$ 判决线,由此算出 z_k 及判决水平或检测水平 y_k 值,y_0 及 σ 由实验求出

$$y_k = y_0 + z_k\sigma_y$$

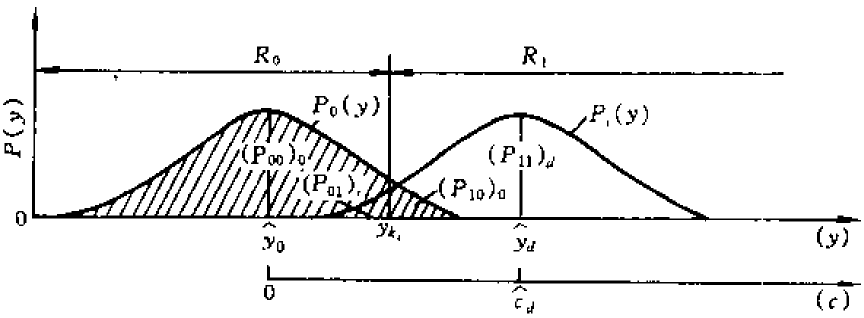


图 5-2 分析方法检测限的定义

在 y_k 确定后,从不同均值 y 对应的分布曲线中,选取其与 $y = y_k$ 判决直线相交给出的 P_{11} 等于 $(P_{11})_d$ 的曲线,这条曲线的均值 y_k 所对应的 c_d 即为所求的浓度检测限。不用图解法亦可由检测特性函数关系直接计算与 P_{11} 对应的 c,亦即检测限 c_d。显然,这样定义的检测限,明确地给定了正确检出的概率 P_{11} 和错误检出的概率 P_{10},而 $P_{00} = 1 - P_{10}$,$P_{01} = 1 - P_{11}$,这样,前面所述的 4 种概率都是完全确定的。分析化学文献中一般定义检测限时所称"可靠地检出"、"以一定的确定度检出"等概念在这里有了明确的数量界限。

检测限的另一种表示方法是给定 P_{11} 及 P_{10} 下对应的 r,或记为 r_d

$$r_d = (y_d - y_0)/\sigma = z_d - z_k$$

r_d 实际就是与检测限对应的信噪比,或称可靠地检出待测组分所需的最低信噪比。噪声是一客观存在的事实,在给定的噪声下,信噪比完全由待测物浓度大小决定。如信噪比低于 r_d,检出将不可能按给定的 P_{11}、P_{10} 值实现。

与检测限 r_d 对应的分析信号 y_d 是

$$y_d = y_0 + r_d\sigma = y_0 + (z_d - z_k)\sigma$$

分析化学文献中一般写成

$$y_d = y_0 + K\sigma$$

有关 K 值的确定，长期未取得一致意见。由上述分析可见，K 的物理意义即是与检测限对应的信噪比。而无论是检测限或信噪比的数值，除与分析方法本身的特性如 y_0（噪声均值）及 σ（标准差）有关外，还取决于我们要求的 P_{11}、P_{10}及分析系统的检测特性函数。一般认为 K 应取 3，可以证明，此时当 P_{10}取足够小（0.001），即犯第一类错误的概率很小时，但 P_{11}仅 0.50，有 50%的概率犯第二类错误。也就是说，K 的确定与我们对分析问题的要求有关，如果要求犯第一类错误（以假为真）的概率要小，但对犯第二类错误（以真为假）的概率无严格要求时，K 可以取小一些；反之，如果要求犯第二类错误的概率亦必需小时，K 值则必须足够大。

第二节　分析数据的统计检验方法

一、分析结果的 t 检验

在分析测试中，测定值将不可避免地要受到许多随机因素的影响，这些个别的因素对测定误差的影响虽然很小，但各个因素的影响综合起来仍会对测定结果产生明显影响。从概率论的观点来看，测定误差是很多数值微小而又相互独立或弱相关的随机变量之总和，是一个随机变量，遵循或近似地遵循正态分布，可用正态分布的概率密度函数来描述大样本测定的误差分布，即

$$x = x_{\text{true}} + e = kc_{\text{true}} + e$$

式中，x 为分析测定数据；x_{true}设为该分析样本在无任何量测误差存在时的真实值，它一般由某一仪器常数与该样本的真实浓度的乘积（kc_{true}）表出；e 为遵循或近似地遵循零均等方差正态分布的量测误差。然而，通常的化学分析测定，都是小样本试验，只进行 3～5 次测定，如此小样本试验并不能求得正态分布的总体均值 μ 和总体标准差 σ，只能求得样本均值$\bar{x}$和样本标准差 s，因此，无法直接应用正态分布于小样本试验数据。直接采用样本均值$\bar{x}$和样本标准差 s 来进行检验，需要应用类似正态分布的 t 分布（参见第十一章）。

1. t 检验的程序

进行 t 检验的程序可归纳如下：

（1）根据实际问题作出原假设 H_0 和备择假设 H_1。

（2）选定所用的统计量：

① 当检验样本均值$\bar{x}$与总体均值 μ 是否存在显著性差异时，使用如下统计量：

$$t = (\bar{x} - \mu)/(s/\sqrt{n})$$

式中，n 为测定次数。

② 当检验两个样本均值之间是否存在显著性差异时，使用统计量：

$$t = (\bar{x}_1 - \bar{x}_2)/[\sqrt{\bar{s}^2(1/n_1 + 1/n_2)}]$$

在此

$$\bar{s}^2 = [(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2]/(n_1 + n_2 - 2)$$

（3）选定显著性水平 α，决定拒绝域 ζ。

(4) 随机抽取样本，计算样本平均值$\bar{x}$和样本方差 S^2。

(5) 计算统计量值。若由样本值计算的统计量落入拒绝域ζ内，即由样本值计算的统计量大于 t 分布表中相应的显著性水平 α 和相应自由度 $f(=n-1)$下的临界值 $t_{\alpha,f}$，则拒绝接受原假设 H_0；若由样本值计算的统计量落入非拒绝域内，即由样本值计算的统计量小于 t 分布表中相应的显著性水平 α 和相应自由度 $f(=n-1)$下的临界值 $t_{\alpha,f}$，则接受原假设 H_0；若由样本值计算的统计量落入拒绝域与非拒绝域的边界附近，即由样本值计算的统计量与 t 分布表中相应的显著性水平 α 和相应自由度 $f(=n-1)$下的临界值 $t_{\alpha,f}$相近似，则原假设 H_0 处于被怀疑状态，这时最好是继续进行试验，而后将新旧样本值合起来计算统计量再进行检验。

2. 举例来说明如何进行 t 检验

某化工厂从以往的生产数据知，在生产正常的情况下，产品中平均含铅量为 0.45%，某日随机抽查了五批产品，测定含铅量分别为 0.48%、0.44%、0.42%、0.47%、0.43%，问在显著性水平 $\alpha=0.05$ 下产品中的平均含铅量有无变化？

解：(1) 原假设 H_0：$\mu=0.45$；备择假设 H_1：$\mu\neq0.45$；

(2) 选定检验统计量

$$t=(\bar{x}-\mu)/(s/\sqrt{n})$$

(3) 选定显著性水平 $\alpha=0.05$，注意到本例为双边检验，即含铅量有无变化（大了与小了都不同于原先的含铅量），查 t 分布表得，$t_{0.05,4}=2.78$，$-t_{0.05,4}=2.78$，则拒绝域为$|t|>2.78$的区域ζ。

(4) 计算样本平均值$\bar{x}$和样本方差 S^2。

$$\bar{x} = (1/n)\sum x_i = 0.448$$

$$S^2 = [1/(n-1)]\sum(x_i-\bar{x})^2 = 0.00067$$

(5) 计算统计量值

$$t=(0.448-0.45)/(s/\sqrt{5})=-0.02/0.0116=-1.727$$

(6) 检验比较。因$|t|=1.727\leqslant2.78$，故可接受原假设，认为测定含铅量与原平均含铅量无显著性变化。

二、分析结果的方差检验

对一个试样进行多次重复测定时，由于测定受到各种因素的影响，各次测定值并不相同，它们之间的差异称为变差。变差的大小通常用样本方差或样本标准差来量度。方差的大小反映了测定结果的精密度，是衡量试验条件稳定性的重要标志。为评价分析方法与测定结果的优劣，比较各种分析方法与测定结果的测定精密度，一般就需采用方差检验的方法。

（一）一个总体的方差检验

若 X 是遵从正态分布 $N(\mu,\sigma^2)$的随机变量，其中 μ 和 σ^2 未知，x_1，x_2，…，x_n 是由总体中随机抽取的样本值，其样本方差为 $S^2(=\sum(x_i-\bar{x})^2/(n-1))$，则$(n-1)S^2/\sigma^2$ 服从 χ^2 分布(见第十章)，即

$$(n-1)S^2/\sigma^2 = \sum((x_i-\bar{x})/\sigma)^2 \sim \chi^2(n-1)$$

其中$(n-1)$为 χ^2 分布的自由度。

对于给定的显著性水平 α，在 χ^2 分布表中可查得自由度 $f=(n-1)$的两侧临界值 $\chi^2_{\alpha/2,f}$和 $\chi^2_{(1-\alpha/2),f}$，使得

$$P\{\chi^2 \geqslant \chi^2_{\alpha/2,f}\} = \alpha/2$$

$$P\{\chi^2 \geqslant \chi^2_{(1-\alpha/2)),f}\} = 1-\alpha/2$$

即
$$P\{\chi^2_{(1-\alpha/2)),f} \leqslant \chi^2 \leqslant \chi^2_{\alpha/2,f}\} = 1-\alpha$$

归结起来，方差检验的一般程序可如下：

(1) 建立原假设 H_0：$S^2=\sigma_0^2$，这里 σ_0 为一要求的方差。

(2) 计算统计量：$\chi^2=(n-1)S^2/\sigma_0^2$。

(3) 给定显著性水平 α，查表得到相应的 $\chi^2_{1-\alpha/2,f}$和 $\chi^2_{\alpha/2,f}$并检验下式是否成立：

$$\chi^2_{1-\alpha/2,f} \leqslant \chi^2 \leqslant \chi^2_{\alpha/2,f}$$

如果成立，则说明原假设成立，否则表明发生了小概率事件，应拒绝原假设。

下面仍用一个实例来加以具体说明。

某钢厂测定铁水中的含碳量，在稳定生产情况下，可假定服从正态分布，其方差按往常冷料可定为 0.048^2，某日测定 5 炉铁水，测得的含碳量分别为 4.32，4.55，4.36，4.40，4.44。试问其方差与 0.048^2 是否存在显著性（显著性水平 $\alpha=0.10$）差异。

① 建立原假设 H_0：$S^2=\sigma_0^2=0.048^2$。

② 计算样本统计量：

$$\chi^2_{样本} = (n-1)S^2/\sigma_0^2 = \sum(x_i-\bar{x})/\sigma^2$$
$$=1/0.048^2[(4.32-4.414)^2+(4.55-4.414)^2+(4.36-4.414)^2+(4.40-4.414)^2+(4.44-4.414)^2]$$
$$=1/0.048^2[(0.094)^2+(0.136)^2+(0.054)^2+(0.014)^2+(0.026)^2]$$
$$=13.6$$

③ 给定显著性水平 $\alpha=0.10$，由 $\chi^2_{\alpha/2,4}=9.49$，$\chi^2_{1-\alpha/2,4}=0.711$，使得

$$P\{0.711 \leqslant \chi^2_{样本} \leqslant 9.49\}=0.90$$

由于 13.6 不在区间 [0.711，9.49] 之内，因而否定 H_0，认为方差显著性改变了，铁水中的碳含量不稳定。

（二）两个总体方差检验

若 x_1，x_2，…，x_{n1}为总体 $N(\mu_1,\sigma_1^2)$的 n_1 个样本，y_1，y_2，…，y_{n2}为总体 $N(\mu_1,\sigma_2^2)$的n_2 个样本，且它们相互独立，由于：

$$(n_1-1)S_1^2/\sigma_1^2 \sim \chi^2(n_{1-1})$$

$$(n_2-1)S_2^2/\sigma_2^2 \sim \chi^2(n_{2-1})$$

我们有

$$(S_1^2/\sigma_1^2)/(S_2^2/\sigma_2^2) \sim F_{(n_1-1,n_2-1)}$$

即：S_1^2/S_2^2 服从第一自由度为 (n_1-1)，第二自由度为 (n_2-1) 的 F 分布（参见第十一章），若假设 H_0：$\sigma_1^2=\sigma_2^2$ 成立，则

$$S_1^2/S_2^2 \sim F_{(n_1-1,n_2-1)}$$

即 S_1^2/S_2^2 服从第一自由度为 (n_1-1) 第二自由度为 (n_2-1) 的 F 分布，且不带未知参数。因此

$$F_{样本}=S_1^2/S_2^2$$

就可作为原假设 H_0 的检验统计量。由上式作为检验统计量进行显著性检验的方法，就称为 F 检验。

F 检验的原理，可以这样来理解。因为样本方差 S_1^2 和 S_2^2 分别是总体方差 σ_1^2 和 σ_2^2 的无偏估计，S_1^2 和 S_2^2 理应分别在 σ_1^2 和 σ_2^2 附近摆动。因此，如果原假设 $\sigma_1^2=\sigma_2^2$ 成立，则 $F_{样本}=S_1^2/S_2^2$ 太大或太小，都表明原假设 H_0 不正确，应拒绝原假设。

现仍以一个实例对 F 检验方法加以具体介绍。

设用两种不同方法冶炼某种金属材料，分别抽样测定其杂质的含量（%），得表 5-1 所示的结果。

表 5-1　两种不同冶炼方法所得的杂质含量

原冶炼方法	新冶炼方法	原冶炼方法	新冶炼方法
26.9	22.6	26.4	23.4
22.8	22.5	27.2	20.5
25.7	20.6	30.2	
23.0	23.6	21.5	
22.3	24.3	29.5	
24.2	21.9	25.1	
26.1	23.2		

问这两种冶炼方法所得的杂质含量的方差是否存在显著性差异。

将原方法和新方法的杂质含量分别计为 ζ 和 η，假定 ζ 服从正态分布 $N(\mu_1,\sigma_1^2)$，η 服从正态分布 $N(\mu_2,\sigma_2^2)$，根据所提问题，提出如下原假设 $H_0:\sigma_1^2=\sigma_2^2$

根据所得数据计算 F 检验统计量：$F_{样本}=S_1/S_2$

$$n_1=13 \quad \bar{x}=25.68 \quad S_1^2=\frac{1}{12}\sum_{i=1}^{13}(x_i-\bar{x})^2=6.178$$

$$n_2=9 \quad \bar{y}=22.51 \quad S_2^2=\frac{1}{8}\sum_{j=1}^{9}(y_j-\bar{y})^2=1.652$$

$$F_{样本(12,8)}=S_1^2/S_2^2=6.178/1.652=3.74$$

对于给定的显著性水平 $\alpha=0.05$，在 F 分布表中查第一自由度为 12，第二自由度为 8 的临界值 $F_{(12,8)}=3.28$，由于 $3.74>3.28$，因而拒绝原假设 H_0，认为这两种冶炼方法的杂质含量的方差是显著不同的。

在此，值得提出的是，在编制 F 分布表时，是将大方差作分子，小方差作分母，所以在由样本值计算统计量 $F_{样本}$时，也要将样本方差 S_1 和 S_2 中数值较大的一个作分子，较小的一个作分母。

第三节　分析信号的平滑方法

平滑方法是分析信号预处理中一种常用的去噪声以提高信噪比的方法，在分析化学的信号预处理中得到了广泛的应用。本节所介绍的平滑方法包括最简单的窗口移动平均法、窗口移动多项式最小二乘拟合平滑法、窗口移动中位数稳健平滑法、粗糙惩罚平滑法。

一、窗口移动平均法

因为对一组含噪声的量测数据进行均值运算时可显著地提高量测数据的信噪比，故此方法可用来对分析信号进行平滑处理，是平滑处理中一个最简单方法。

对于一个含噪声的波谱或电化学谱（其量测点为 n 个），若对其中第 i 个点及其前后各

m 个数据点进行平均，使得参加平均的数据点一共为 $N=2m+1$，则窗口移动平均法的计算公式为

$$x_{i,平滑后}=\frac{1}{N}\sum_{j=-m}^{m}x_{i+j,平滑前}(i=1,\cdots,n)$$

明显可知平滑后的第 i 个点的数值为这（$2m+1$）个量测数据的平均值，（$2m+1$）也就是窗口的宽度。然后依次移动此窗口，以上式求得第（$i-1$）个点上的平滑后数据，直至将整个波谱量测点数据计算完，故此法有窗口移动平均法之称。图 5-3 对此给出了一个形象示意图[8]。

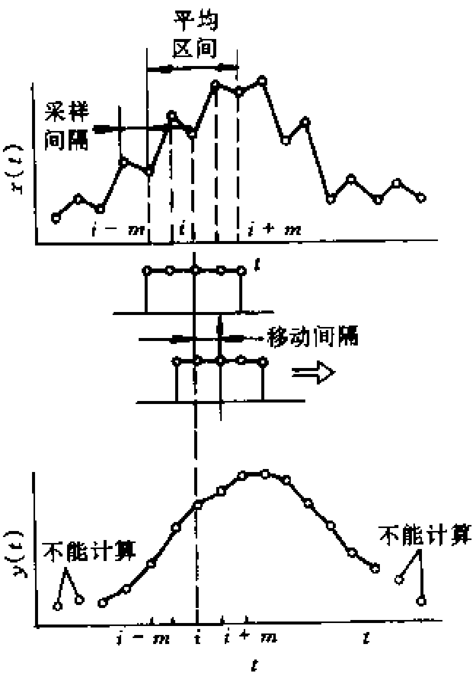


图 5-3 窗口移动平均法的示意图

从图 5-3 中还可以看出，采用这种窗口移动法对波谱量测值前 m 个点，以及后 m 个点不能进行平滑运算，故所得的平滑后波谱会有丢失这些边界点信息的损失。另外，窗口宽度（$2m-1$）的选择也是一个值得讨论的重要因素，窗口太小时（如 $2m+1=3$），平滑去噪效果将不佳，而窗口太大时，则由于此法是进行简单求均值运算，平滑后的波谱有波形失真之弊。这些因素在选用此法进行平滑处理时，都是值得考虑的因素。

二、窗口移动多项式最小二乘拟合平滑法

在此介绍的窗口移动多项式最小二乘平滑法是由 Savitzky 与 Golay[8]共同提出的，此法不是采用简单平均运算，而是在窗口移动运算中引入多项式最小二乘拟合，故该法可做到既能去噪提高分析信号的信噪比，又可较好地保持分析信号（波谱）中的有用信息。此法虽早在 20 世纪 60 年代提出，至今仍在分析化学信号预处理中频繁使用。

窗口移动多项式最小二乘拟合法计算的基本思路与前述窗口移动平均是类似的，唯一不同的地方只是此法没有使用简单的平均而是通过多项式来对移动窗口之内的波谱点进行多项式最小二乘拟合，将窗口内 $N=2m+1$ 个等距离点（m 一般可取 1 至 12）的数据，拟合为（$k-1$）阶多项式

$$x_j^i=a_0+a_1j+a_2j^2+\cdots+a_kj^k\quad(j=-m,-m+1,\cdots,m-1,m,i=1,\cdots,n)\tag{5-1}$$

通过对上式进行最小二乘拟合就可得到一个求得 x_0^i，即窗口内中心点的计算公式，以后我们就可以看到，所得的公式与窗口移动平均法的唯一不同点就在于多项式拟合法实质上是一种加权平均法，它更强调中心点的中心作用而已。

下面以一个窗口为 5 的二次多项式为例来加以具体说明。根据式（5-1）有

$$\begin{cases}x_{-2}^{i-2}=a_0+a_1(-2)+a_2(-2)^2 & (5\text{-}2a)\\ x_{-1}^{i-1}=a_0+a_1(-1)+a_2(-1)^2 & (5\text{-}2b)\\ x_0^i=a_0+a_1(0)+a_2(0)^2 & (5\text{-}2c)\\ x_1^{i+1}=a_0+a_1(1)+a_2(1)^2 & (5\text{-}2d)\\ x_2^{i+2}=a_0+a_1(2)+a_2(2)^2 & (5\text{-}2e)\end{cases}$$

亦即

$$\begin{cases} x_{-2}^{i-2}=a_0-2a_1+4a_2 \\ x_{-1}^{i-1}=a_0-a_1+a_2 \\ x_0^i=a_0 \\ x_1^{i+1}=a_0+a_1+a_2 \\ x_2^{i+2}=a_0+2a_1+4a_2 \end{cases} = \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \tag{5-3}$$

$$\boldsymbol{x}=\boldsymbol{Ma}$$

其中

$$\boldsymbol{x}=\begin{bmatrix} x_{-2}^{i-2} \\ x_{-1}^{i-1} \\ x_0^i \\ x_1^{i+1} \\ x_2^{i+2} \end{bmatrix} \tag{5-4}$$

$$\boldsymbol{M}=\begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \quad \boldsymbol{a}=\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \tag{5-5}$$

解此方程可得

$$\hat{\boldsymbol{a}}=(\boldsymbol{M}^t\boldsymbol{M})^{-1}\boldsymbol{M}^t\boldsymbol{x}$$

将所求$\hat{\boldsymbol{a}}$代入式$\hat{\boldsymbol{x}}=\boldsymbol{M}\,\hat{\boldsymbol{a}}=\boldsymbol{M}(\boldsymbol{M}^t\boldsymbol{M})^{-1}\boldsymbol{M}^t\,\boldsymbol{x}$中，即可得

$$\hat{x}_{-2}^{i-2}=\frac{1}{35}(31x_{-2}^{i-2}+9x_{-1}^{i-1}-3x_0^i-5x_1^{i+1}+3x_2^{i+2}) \tag{5-6a}$$

$$\hat{x}_{-1}^{i-1}=\frac{1}{35}(9x_{-2}^{i-2}+13x_{-1}^{i-1}+12x_0^i+6x_1^{i+1}-5x_2^{i+2}) \tag{5-6b}$$

$$\hat{x}_0^i=\frac{1}{35}(-3x_{-2}^{i-2}+12x_{-1}^{i-1}+17x_0^i+12x_1^{i+1}-3x_2^{i+2}) \tag{5-6c}$$

$$\hat{x}_1^{i+1}=\frac{1}{35}(-5x_{-2}^{i-2}+6x_{-1}^{i-1}+12x_0^i+13x_1^{i+1}+9x_2^{i+2}) \tag{5-6d}$$

$$\hat{x}_2^{i+2}=\frac{1}{35}(3x_{-2}^{i-2}-5x_{-1}^{i-1}-3x_0^i+9x_1^{i+1}+3x_2^{i+2}) \tag{5-6e}$$

在上述方程中，$\hat{x}_j^i(j=-2,-1,0,1,2),(i=1,\cdots n)$是估计值$x_j^i$计算而得，其上标$i$为波谱量测点的位置（紫外光谱为波长，红外光谱为波数，电化学谱则为电位或其他变量离散点）。而下标j则表示在窗口中相对于中心点0的位置点，故有正负之别，负数表示在中心点0位置的左边，正数表示在中心点的右边。

如前所述，窗口移动多项式最小二乘拟合法与窗口移动平均法的差别仅在于前者是一种强调中心点作用的加权平均，故在上述式(5-6)5个式子中，窗口移动多项式最小二乘拟合法真正用到的只是中心点，即$\hat{x}_0^i$的表达式

$$\hat{x}_0^i=\frac{1}{35}(-3x_{-2}^{i-2}+12x_{-1}^{i-1}+17x_0^{i+1}+12x_1^{i+1}-3x_2^{i+2}) \tag{5-7}$$

除去移动窗口位置下标，并将其整理成如 $x_{i,平滑后}=\frac{1}{N}\sum_{j=-m}^{m}x_{i+j,平滑前}$ 相类似式子后得

$$x_i=\frac{1}{A}\sum_{j=-m}^{m}w_j x_{i+j} \tag{5-8}$$

式（5-8）可称为窗口移动多项式最小二乘拟合法的通式，式中 A 可称为归一化常数，$(2m+1)$ 为窗口宽度。w_j 为相应的权重数，故在窗口宽度确定以后，就有其相应的 w_j。值得指出的是多项式次数的选择是可变的，前述例子只是给出了二次多项式平滑的例子。实际亦可用 3 次，4 次甚至更高次的多项式来拟合。当然相应的归一化常数和权重就会发生变化了。为此 Savitzky 和 Golay 就计算出了一系列不同权重与归一化常数，以表的形式给出使读者可直接使用它们来进行平滑计算。显然，$w_j(j=-m,\cdots m)$的个数也就是窗口的宽度。为方便读者直接进行平滑，我们将它们收录于表 5-2 和表 5-3 之中。

在 Savitzky 和 Golay 给出的表中存在一些错误，后由 J. Steinier 等对其进行了修正[10]，并由 P. A. Gorry 给出了更一般的形式和计算边界点公式[illegible]，本书给出的表 5-2 及表 5-3 是经过修正的。

表 5-2　窗口移动多项式拟合平滑权重系数表（多项式次数：2 或 3）

窗口宽度 $(N=2m+1)$	25	23	21	19	17	15	13	11	9	7	5
−12	−253										
−11	−138	−42									
−10	−33	−21	−171								
−09	62	−2	−76	−136							
−08	147	15	9	51	−21						
−07	222	30	84	24	−6	−78					
−06	287	13	149	89	7	13	11				
05	343	54	204	144	18	42	0	−36			
−04	387	63	249	189	27	87	9	9	−21		
−03	422	70	284	224	34	122	16	44	14	−2	
−02	447	75	309	249	39	147	21	69	39	3	−3
−01	462	78	324	264	42	162	21	84	54	6	12
00	467	79	329	269	43	167	25	89	59	7	17
01	462	78	324	264	42	162	21	84	54	6	12
02	447	75	309	249	39	147	21	69	39	3	−3
03	422	70	284	224	34	122	16	44	14	−2	
04	387	63	249	189	27	87	9	9	−21		
05	343	54	204	144	18	42	0	−36			
06	287	43	149	89	7	−13	−11				
07	222	30	84	24	−6	−78					
08	147	15	9	−51	−21						
09	62	−2	−76	−136							
10	−33	−21	−171								
11	−138	−42									
12	−253										
归一化常数	5175	805	3059	2261	323	1105	143	429	231	21	35

表 5-3　窗口移动多项式拟合平滑权重系数表（多项式次数：4 或 5）

窗口宽度 ($N=2m+1$)	25	23	21	19	17	15	13	11	9	7
−12	1265									
−11	−345	95								
−10	−1122	−38	11628							
−09	−1255	−95	−6460	340						
−08	−915	−95	−13005	−255	195					
07	−255	−55	−11220	−420	−195	2145				
−06	590	10	−3940	−290	−260	−2860	110			
−05	1503	87	6378	18	−117	2937	−198	18		
−04	2385	165	17655	405	135	−165	−135	−45	15	
−03	3155	235	28190	790	415	3755	110	−10	−55	5
−02	3750	290	36660	1110	660	7500	390	60	30	−30
01	4125	325	42120	1320	825	10125	600	120	135	75
00	4253	−337	44003	1393	883	11063	677	143	179	131
01	4125	325	42120	1320	825	10125	600	120	135	75
02	3750	290	36660	1110	660	7500	390	60	30	−30
03	3155	235	28190	790	415	3755	110	−10	−55	5
04	2385	165	17655	405	135	−165	−135	−45	15	
05	1503	87	6378	18	−117	−2937	−198	18		
06	590	10	−3940	−290	−260	−2860	110			
07	−255	−55	−11220	−420	−195	2145				
08	−915	−95	−13005	−255	195					
09	−1255	−95	−6460	340						
10	−1122	−38	11628							
11	−345	95								
12	1265									
归一化常数	30015	6555	260015	7429	4199	16189	2431	429	429	231

三、窗口移动中位数稳健平滑法

由于平均数的计算和最小二乘估计是非稳健的（non-robust）（参见第五章），所以对于一些存在有奇异点的波谱，前述两种平滑法皆难于给出满意的结果。如果用稳健的中位数来代替平均数，就构成了窗口移动中位数稳健平滑法，此法的基本计算过程与窗口移动平均法完全一样。只需将其计算公式作一改动即可。

$$x_{i,\text{平滑后}}=\text{median}(x_{i+j}) \qquad j=(-m,\cdots,m)$$

此法虽然简单，但仍不失为一个可用的稳健平滑方法。

四、粗糙惩罚平滑法

在前述几小节讨论的平滑方法中，窗口移动多项式最小二乘法在化学波谱数据预处理中得到了相当广泛的应用，确为一个较好的平滑方法。然而，该法也存在一些弱点：① 该法是一种基于最小二乘拟合的方法，很不稳健，其崩溃点为 $1/n$（参见第五章稳健估计），故对于存在异常点（outlier）的数据，所得结果将不尽人意。② 该法本质上是一种参数估计法，其结果与适当参数选取密切相关，其中窗口的宽度，多项式的次数的选取就十分重要，不同的参数得到的结果就不尽相同。③ 由于该法实质上采用中心点来进行加权估计，所以有边界点信

息丢失之虞。窗口越大，丢失的点就越多。虽提出了一个边界点计算方法，但其应用远不及原始 Savitzky 和 Golay 法广泛。所以，发展更有效的方法仍有必要。

本节所讨论的粗糙惩罚平滑法（roughness penalty smoothing method）是一种非参数估计的方法，故可克服前述 Savitzky 法的缺点。该法是由英国统计学家 Silverman 提出，其基本思路是在最小二乘估计的基础上再加上一个粗糙惩罚项，其目标函数为

$$S[f(w)] = \sum_{i=1}^{n}(x_i - f_i)^2 + \alpha\int_{w_1}^{w_n}[f''(w)]^2\mathrm{d}w \tag{5-9}$$

式中，f 是一个任意二阶可微函数，$x_i(i=1,\cdots,n)$为在某波谱（包括电化学谱）中的量测点数据，f_i 或 $f(w_i)$为 $f(w)$在 $w_i(i=1,\cdots,n)$上的函数值，$f''(w)$为其二阶导数，α 为惩罚参数，其大小决定了惩罚项在目标函数中的作用。其外，此目标函数用了 f 的二阶导数积分值作为其惩罚项，实因二阶导数之积分值是任一曲线粗糙性的最好度量，其与样条的张力能（strain energy）是成正比的。

从上述目标函数可以看出，粗糙惩罚平滑法一方面注意到了拟合的作用（此点优于简单平均），另一方面，又注意到了测量数据的粗糙性，越是粗糙和不光滑的测量数据中噪音的成分就越大，受到的惩罚亦越大。通过交叉校验（cross-validation）来决定 α，则可使该目标函数在对拟合噪声惩罚达到最好的平衡，以最大限度保证分析信号既不失真又能除噪，取得最好的平滑效果。在此，必须指出的另一点是，由于 f 为任一二阶可微函数，所以此目标函数为求一泛函最优问题。

正由于这问题为一泛函寻优问题，一般很难用经典方法求解，故此法的求解方式是首先转换成矩阵形式，继续将其变形。从而得到它的最优解。

首先，可以证明

$$\int_{w_1}^{w_n}[f''(w)]^2\mathrm{d}w = \boldsymbol{f}'\boldsymbol{K}\boldsymbol{f} \tag{5-10}$$

其中，$\boldsymbol{K}$ 为一$(n\times n)$阶矩阵，$\boldsymbol{f}=[f_1,\cdots,f_n]^{\mathrm{t}}$，既所求函数在 n 个样点上的函数值。

$\boldsymbol{K}$ 矩阵可由下列方法来构成：

首先令 $h_i = w_{i+1} - w_i\ (i=1,\cdots,n-1)$，继先构成两个矩阵，即矩阵 $\boldsymbol{Q}_{n\times(n-2)}$ 和矩阵 $\boldsymbol{R}_{(n-2)\times(n-2)}$其中矩阵 $\boldsymbol{Q}$ 的元素可由下式定义

$$q_{j-1,j}=1/h_{j-1} \quad ; \quad q_{jj}=-1/h_{j-1}-1/h_j \quad ; \quad q_{j+1,j}=1/h_j \tag{5-11}$$

$$j=2,\cdots,n-1$$

而对于一般的元素 q_{ij}当$|i-j|\geqslant 2$ 时，$q_{ij}=0(i=1,\cdots,n;j=1,\cdots,n-1)$

所以有

$$\boldsymbol{Q}=\begin{bmatrix} 1/h_1 & 0 & 0 & \cdots & 0 \\ 1/h_1-1/h_2 & 1/h_2 & 0 & \cdots & 0 \\ 1/h_2 & 1/h_2-1/h_3 & 1/h_3 & & \\ 0 & 1/h_3 & 1/h_3-1/h_4 & & \vdots \\ 0 & 0 & 1/h_4 & & \\ & & 0 & & \\ \vdots & \vdots & \vdots & & 1/h_{n-2} \\ 0 & 0 & 0 & 1/h_{n-2} & 1/h_{n-1}-1/h_{n-2} \\ 0 & 0 & 0 & 0 & 1/h_{n-1} \end{bmatrix}_{n\times(n-2)}$$

矩阵 $\boldsymbol{R}$ 则为一个$(n-2)\times(n-2)$对称方阵，其主要非零元素

$$r_{ii}=(h_{i-1}+h_i)/3 \qquad (i=2,\cdots,n-1); \tag{5-12}$$

$$r_{i,i+1}=r_{i+1,i}=h_i/6 \qquad (i=2,\cdots,n\text{-}1) \tag{5-13}$$

而当$|i-j|\geqslant 2$时，$r_{ij}=0$，即

$$\boldsymbol{R}=\begin{vmatrix} (h_1+h_2)/3 & h_2/6 & 0 & \cdots & 0 & 0 \\ h_2/6 & (h_2+h_3)/3 & h_3/6 & \cdots & 0 & 0 \\ 0 & h_3/6 & (h_3+h_4)/3 & \cdots & 0 & 0 \\ 0 & 0 & h_4/6 & \cdots & 0 & \\ 0 & 0 & 0 & \cdots & \vdots & \vdots \\ & & 0 & \cdots & & \\ \vdots & \vdots & 0 & \cdots & 0 & \\ & & \vdots & & h_{n-3}/6 & 0 \\ & & & & (h_{n-3}+h_{n-2})/3 & h_{n-2}/6 \\ 0 & 0 & 0 & \cdots & h_{n-2}/6 & (h_{n-2}+h_{n-1})/3 \end{vmatrix}_{(n-2)\times(n-2)}$$

如果所需平滑的波谱(包括电化学谱)是等距(等时)采样的，则上述两个矩阵 $\boldsymbol{Q}$ 和矩阵 $\boldsymbol{R}$ 就可简化为

$$\boldsymbol{Q}=\begin{vmatrix} 1 & 0 & 0 & \cdots & 0 \\ -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & \vdots \\ & & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & 1 \\ & & & & -2 \\ 0 & 0 & 0 & \cdots & 1 \end{vmatrix}_{n\times(n-2)}$$

$$\boldsymbol{R}=\begin{vmatrix} 2/3 & 1/6 & 0 & \cdots & 0 & 0 \\ 1/6 & 2/3 & 1/6 & \cdots & 0 & 0 \\ 0 & 1/6 & 2/3 & \cdots & 0 & 0 \\ 0 & 0 & 1/6 & \cdots & 0 & \\ 0 & 0 & 0 & \cdots & \vdots & \vdots \\ & & 0 & \cdots & & \\ & & 0 & \cdots & 0 & \\ \vdots & \vdots & \vdots & & 1/6 & 0 \\ & & & & 2/3 & 1/6 \\ 0 & 0 & 0 & \cdots & 1/6 & 2/3 \end{vmatrix}_{(n-2)\times(n-2)}$$

如果 $f(w)$可由一个自然立方样条(natural cubic spline)函数表出，而且在每一样点 $w_1<\cdots<w_n$ 上，下式成立，即

$$f_i=f(w_i) \qquad \gamma_i=f''(w_i) \qquad (i=1,2,\cdots,n)$$

而且，该函数在边界点，即 w_1 和 w_n 上的二阶和三阶导数值皆为零，则下而的引理成立。

引理：矢量 $f(w)$ 和 γ 定义了一个自然立方样条函数 $f(w)$，当且仅当下述条件：

$$Q^t f = R\gamma \tag{5-14}$$

成立。如式（5-14）成立，则惩罚项可满足下式：

$$\int_{w_1}^{w_n}[f''(w)]^2 dw = \gamma^t R \gamma = f^t K f \tag{5-15}$$

此引理的证明可参阅文献[12]

利用此引理，可容易地求出矩阵 K 的表达式来。因为

$$\gamma = R^{-1} Q^t f \tag{5-16}$$

$$\gamma^t = f^t Q^t (R^{-1})^t = f^t Q^t R^{-1} \tag{5-17}$$

实因 R 为一对称方阵。对照前述(5-16)及(5-17)式可得

$$K = Q R^{-1} Q^t \tag{5-18}$$

由于求出了 K 矩阵的表达式，可将粗糙惩罚平滑法的目标函数式用矩阵形式表达如下：

$$\begin{aligned} S[f(w)] &= \sum_{i=1}^{n}(x_i - f_i)^2 + \alpha\int_{w_1}^{w_n}[f''(w)]^2 dw \\ &= (x - f)^t(x - f) + \alpha f^t K f \\ &= x^t x - 2f^t x + f^t(I + \alpha K) f \end{aligned} \tag{5-19}$$

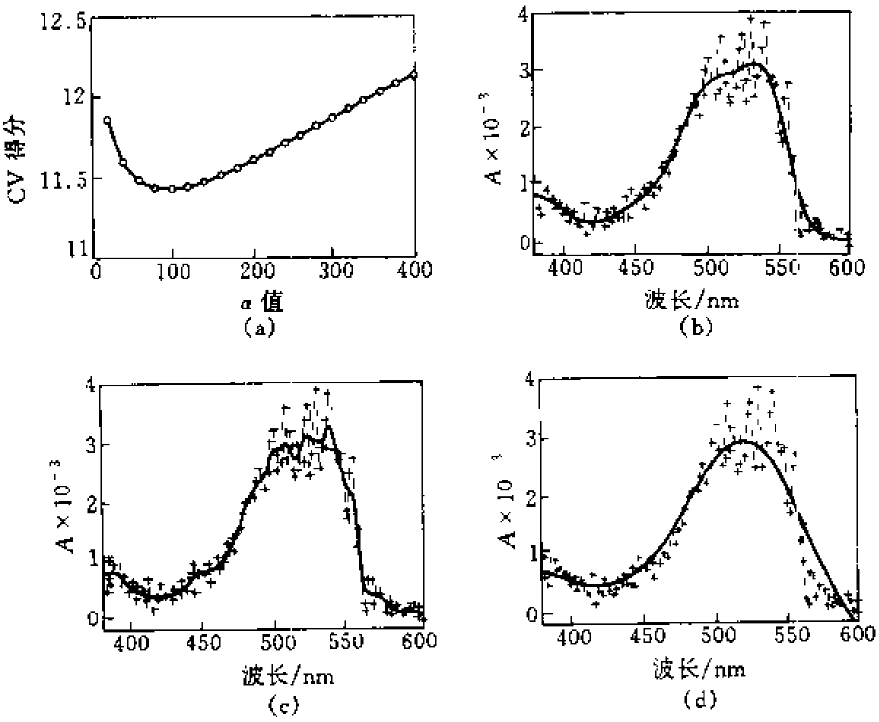


图 5-4　粗糙惩罚平滑法处理的一个实例

(a) 交叉校验法得分曲线。$\alpha=90$ 时取得最小值；(b) $\alpha=90$ 时粗糙惩罚平滑结果；

(c) $\alpha=30$ 时粗糙平滑结果；(d) $\alpha=300$ 时粗糙平滑结果

矩阵 $(I+\alpha K)$ 为一严格正定矩阵，这说明上式具有唯一的极小值，而且使式（5-19）成立的极小值的解为

$$f = (I+\alpha K)^{-1} x \tag{5-20}$$

这是因为，目标函数 $S[f(w)]$ 可表示为

$$\{f-(I+\alpha K)^{-1}x\}^t (I+\alpha K)\{f-(I+\alpha K)^{-1}x\} + S \tag{5-21}$$

其中 S 为一与 f 无关的常数项。从式（5-21）可以清楚看出，只有当 $f=(I+\alpha K)^{-1}x$ 时 $S[f(w)]$，才达到极小。

因 $x=[x_1,\cdots,x_n]^t$ 为已知。K 可根据式(5-18)求出，只要给定了 α，运用MATLAB标准程序是很容易求出 f 来的。

原则上说 α 是可由不同方法确定，本书介绍的是交叉校验法(cross validation)可以证明[12]，如果我们是采用一次剔除一个样本点的算法，其交叉校验的总得分可由下式给出。

$$CV=n^{-1}\sum_{i=1}^{n}\left|\frac{x_i-f_i}{1-A_{ii}(\alpha)}\right|^2 \tag{5-22}$$

式中，$A(\alpha)=(I+\alpha K)^{-1}$，而 $A_{ii}(\alpha)$ $(i=1,\cdots n)$是该矩阵的对角元素；f_i 为所估计的函数值。

有关如何构成此交叉校验法的细节因涉及过多的数学推导，读者可参阅文献[12]。

图 5-4 给出了一个用粗糙惩罚平滑法处理的一个实例。从图 5-4 中可以看出，由交叉校验法给出的 α 的确可以达到既不失真又达到了除噪的最佳选择。

第四节　分析信号的求导方法

在波谱分析中，导数波谱为一重要的方法，如何直接从所得的离散波谱中获得导数波谱一直是分析化学中一个值得研究的同题。本节将对两种常用的求导方法进行介绍。

一、直接差分法

直接差分法是一种最简单的离散波谱求导方法，对于一个离散波谱 $x_i(i=1,\cdots,n)$，其采样波长（w_i）或时间点（t_i）（$i=1$，…，n）来说，则计算公式为

$$y_i=(x_i+1-x_i)/(w_{i+1}-w_i) \tag{5-23}$$

如果采样波长或时间为等距的，则式（5-23）可改写为

$$y_i=x_{i+1}-x_i\quad(i=1,\cdots,n-1) \tag{5-24}$$

在 MATLAB 中，有一个 deriv 函数，就是用式（5-24）来进行计算的。

直接差分法简单，但有两个缺点，一是所求导数谱比原始波谱少了一个采样点，而使其发生了半个点的位移，这对于要使用原始波谱极值点数据时带来误差；二是这种方法对于分辨率高的波谱或是采样时间点很密的波谱所求导数与实际相差不大，但对于一些分辨率低，即采样点稀的波谱，此法所求的变化率（导数）就存在较大误差了，所以在使用 MATLAB 的 deriv 函数时一定要对这两点加以注意。

二、多项式最小二乘拟合求导法

在前一节有关分析信号的平滑方法中，介绍了 Savitzky 和 Golay 的窗口移动多项式最小二乘拟合法，实际上该法不但可用于平滑，亦可用于求导，实因其为一多项式拟合，可以对此多项式求导，就可直接得到求导所需的窗口中心点的加权平均表达式。此法求导所得的导数波谱，能克服前述简单差分法的弱点，且导数谱不发生位移可精确求得各采样点导数。

前述求窗口（$N=2m+1$）中的一个中心点 x_0^i 的表达式，即

$$x_j^{i+j}=a_0+a_1j+a_2j^2+\cdots+a_kj^k(i=1,\cdots,n)(j=-m,\cdots,0,\cdots,m) \tag{5-25}$$

一阶导数：

$$\mathrm{d}(x_j^{i+j})/\mathrm{d}(j)=a_1+2a_2j+\cdots+ka_kj^{k-1} \tag{5-26}$$

当 $j=0$ 时，有

$$\mathrm{d}(x_j^{i+j})/\mathrm{d}(j)_{j=0}=a_1 \tag{5-27}$$

二阶导数：$d^2(x_j^{\prime -\prime})/d(j)=2a_2+\cdots+(k-1)ka_kj^{k-1}$ (5-28)

当 $j=0$ 时，有 $d^2(x_j^{\prime +\prime})/d(j)_{j=0}=2a_2$

如此等等，可一直求得该多项式的 $(k-1)$ 阶导数。窗口移动法实际上是只使用中心点的信息，也就是当 $j=0$ 时的信息，所以有

$$d^k(x_j^{\prime -\prime})/d(j)_{j=0}=k!\ a_k \tag{5-29}$$

由此可知，如果求得了 $a_i(i=1,\cdots,k)$ 就可求出 1 至 k 阶的权重表达式。例如窗口宽度 $N=2m+1=9$，以 4 阶多项拟合得到的 3 阶的权重表达式如式(5-30)所示。

$$x_0^{m''}=(-14x'_{-4}+7x'_{-3}+13x'_{-2}+9x'_{-1}-9x'_{1}-13x'_{2}-7x'_{3}+14x'_{4}) \tag{5-30}$$

表 5-4～表 5-9 是将一些权重系数以表的形式给出，以供读者选择使用。

表 5-4　窗口移动多项式拟合求一阶导数权重系数表（多项式次数 3 或 4）

窗口宽度	25	23	21	19	17	15	13	11	9	7	5
−12	30866										
−11	8602	3938									
−10	−8525	815	84075								
−9	−20982	−1518	10032	6936							
−8	−29236	−3140	−43284	68	748						
−7	−33754	−4130	−78176	−4648	−98	12922					
−6	−35003	−4567	−96947	−7481	−643	−4121	1133				
−5	−33450	−4530	−101900	−8700	−930	−14150	660	300			
−4	−29562	−4098	−95338	−8574	−1002	18331	−1578	−294	86		
−3	−23806	−3350	−79564	−7372	−902	−17842	−1796	−532	−142	22	
−2	−16649	−2365	−56881	−5363	−673	−13843	−1489	503	−193	−67	1
−1	−8558	−1222	−29592	−2816	−358	−7506	−832	−296	−126	−58	−8
0	0	0	0	0	0	0	0	0	0	0	0
1	8558	1222	29592	2816	358	7506	832	296	126	58	8
2	16649	2365	56881	5363	673	13843	1489	503	193	67	−1
3	23806	3350	79564	7372	902	17842	1796	532	142	−22	
4	29562	4098	95338	8574	1002	18334	1578	294	−86		
5	33450	4530	101900	8700	930	14150	660	−300			
6	35003	4567	96947	7481	643	4121	−1133				
7	33754	4130	78176	4648	98	−12922					
8	29236	3140	43284	−68	−748						
9	20982	1518	−10032	−6936							
10	8525	−815	−84075								
11	−8602	−3938									
12	−30866										
归一化常数	1776060	197340	3634092	255816	23256	334152	24024	5148	1188	252	12

表 5-5　窗口移动多项式拟合求一阶导数权重系数表（多项式次数 5 或 6）

窗口宽度	25	23	21	19	17	15	13	11	9	7	5
−12	−8322182										
−11	6024183	−400653									
−10	9604353	359157	−15033066								
−9	6671883	489687	16649358	−255102							
−8	544668	265164	19052988	349928	−14404						
−7	−6301491	−106911	6402438	322378	24661	−78351					

续表

窗口宽度	25	23	21	19	17	15	13	11	9	7	5
−6	−12139321	−478349	−10949942	9473	16679	169819	−9647				
−5	−15896511	−752859	−26040033	348823	−8671	65229	27093	−573			
−4	−17062146	−878634	−24807914	−604484	−32306	−130506	−12	2166	−254		
−3	−15593141	−840937	−35613829	−686099	−43973	−266401	−33511	−1249	1381	−1	
−2	−11820675	−654687	−28754154	−583549	−40483	279975	−45741	−3774	−2269	9	
−1	−6356625	−357045	−15977364	−332684	−23945	−175125	31380	−3084	−2879	−15	
0	0	0	0	0	0	0	0	0	0	0	
1	6356625	357045	15977364	332684	23945	175125	31380	3084	2879	45	
2	11820675	654687	28754154	583549	40483	279975	45741	3774	2269	−9	
3	15593141	840937	35613829	686099	43973	266401	33511	1249	−1381	1	
4	17062146	878634	34807914	604484	32306	130506	12	−2166	254		
5	15896511	752859	26040033	348823	8671	−65229	−27093	573			
6	12139321	478349	10949942	9473	16679	169819	9647				
7	6301491	106911	6402438	−322378	−24661	78351					
8	−544668	−265164	−19052988	−349928	14404						
9	−6671883	−489687	−16649358	255102							
10	−9604353	−359157	15033066								
11	−6024183	400653									
12	8322182										
归一化常数	429214500	18747300	637408200	9806280	503880	2519400	291720	17160	8580	60	

表 5-6　窗口移动多项式拟合求二阶导数权重系数表（多项式次数 2 或 3）

窗口宽度	25	23	21	19	17	15	13	11	9	7	5
−12	92										
−11	69	77									
−10	48	56	190								
−9	29	37	133	51							
−8	12	20	82	34	40						
−7	−3	5	37	19	25	91					
−6	−16	−8	−2	6	12	52	22				
−5	−27	−19	−35	−5	1	19	11	15			
−4	−36	−28	−62	−14	−8	−8	2	6	28		
−3	−43	−35	−83	−21	−15	−29	−5	−1	7	5	
−2	−48	−40	−98	−26	−20	−44	−10	−6	−8	0	2
−1	−51	−43	−107	−29	−23	−53	−13	−9	−17	−3	−1
0	−52	−44	−110	−30	−24	−56	−14	−10	−20	−4	−2
1	−51	−43	−107	−29	−23	−53	−13	−9	−17	−3	−1
2	−48	−40	−98	−26	−20	−44	−10	−6	−8	0	2
3	−43	−35	−83	−21	−15	−29	−5	−1	7	5	
4	−36	−28	−62	−14	−8	−8	2	6	28		
5	−27	−19	−35	−5	1	19	11	15			
6	−16	−8	−2	6	12	52	22				
7	−3	5	37	19	25	91					
8	12	20	82	34	40						
9	29	37	133	51							
10	48	56	190								
11	69	77									
12	92										
归一化常数	26910	17710	33649	6783	3876	6188	1001	429	462	42	7

表 5-7　窗口移动多项式拟合求二阶导数权重系数表（多项式次数 4 或 5）

窗口宽度	25	23	21	19	17	15	13	11	9	7	5
−12	−143198										
−11	10373	−115577									
−10	99385	20615	−12597								
−9	137803	93993	3876	−32028							
−8	138262	119510	11934	15028	−2132						
−7	112067	110545	13804	35148	1443	−31031					
−6	69193	78903	11451	36357	2691	29601	−2211				
−5	18285	34815	6578	25610	2405	44495	2970	−90			
−4	−33342	−13062	626	8792	1256	31856	3504	174	−126		
−3	−79703	−57645	−5226	−9282	−207	6579	1614	146	371	−13	
−2	−116143	−93425	−10061	24867	−1557	19751	−971	1	151	67	
−1	−139337	−116467	−13224	−35288	−2489	−38859	−3016	−136	−211	−19	
0	−149290	124410	−14322	−38940	−2820	−45780	−3780	−190	−370	−70	
1	−139337	−116467	−13224	−35288	−2489	−38859	−3016	−136	−211	−19	
2	−116142	−93425	−10061	−24867	−1557	−19751	−971	1	151	67	
3	−79703	−57645	−5226	−9282	−207	6579	1614	146	371	−13	
4	−33342	−13062	626	8792	1256	31856	3504	174	−126		
5	18285	34815	6578	25610	2405	44495	2970	−90			
6	69193	78903	11451	36357	2691	29601	−2211				
7	112067	110545	13804	35148	1443	−31031					
8	138262	119510	11934	15028	−2132						
9	137803	93993	3876	−32028							
10	99385	20615	−12597								
11	10373	−115577									
12	−134198										
归一化常数	17168580	11248380	980628	1961256	100776	1108536	58344	1716	1716	132	

表 5-8　窗口移动多项式拟合求三阶导数权重系数表（多项式次数 3 或 4）

窗口宽度	25	23	21	19	17	15	13	11	9	7	5
−12	−506										
−11	−253	−77									
−10	−55	−35	285								
−9	93	−3	−114	−204							
−8	196	20	12	−68	−28						
−7	259	35	98	28	−7	−91					
−6	287	43	149	89	7	−13	−11				
−5	285	45	170	120	15	35	0	−30			
−4	258	42	166	126	18	58	6	6	−14		
−3	211	35	142	112	17	61	8	22	7	−1	
−2	145	25	103	83	13	49	7	23	13	1	−1
−1	77	13	54	44	7	27	4	14	9	1	2
0	0	0	0	0	0	0	0	0	0	0	0
1	−77	−13	−54	−44	−7	−27	−4	−14	−9	−1	−2
2	−149	−25	−103	−83	−13	−49	−7	−23	−13	−1	1
3	−211	−35	−142	−112	−17	−61	−8	−22	−7	1	
4	−258	−42	−166	−126	−18	−58	−6	−6	14		
5	−285	−45	−170	−120	−15	−35	0	30			
6	−287	−43	−149	−89	−7	13	11				

续表

窗口宽度	25	23	21	19	17	15	13	11	9	7	5
7	−259	−35	−98	−28	7	91					
8	−196	−20	−12	68	28						
9	−93	3	114	204							
10	55	35	285								
11	253	77									
12	506										
归一化常数	296010	32890	86526	42636	3876	7956	572	858	198	6	2

表 5-9　窗口移动多项式拟合求三阶导数权重系数表（多项式次数 5 或 6）

窗口宽度	25	23	21	19	17	15	13	11	9	7	5
−12	284372										
−11	−144463	49115									
−10	−293128	−32224	748068								
−9	−266403	−55233	−625974	15810							
−8	−146408	−43928	−908004	16796	1144						
−7	5131	−16583	−598094	−20342	−1547	8281					
−6	144616	13632	−62644	−9818	−1508	−14404	1430				
−5	244311	38013	448909	4329	−351	−10379	−3267	129			
−4	290076	51684	787382	15546	876	1916	−1374	−402	100		
−3	279101	52959	887137	20525	1595	11671	1633	−11	−457	1	
−2	217640	42704	749372	18554	1604	14180	3050	340	256	8	
−1	118745	23699	425412	10868	983	9315	2252	316	459	13	
0	0	0	0	0	0	0	0	0	0	0	
1	−118745	−23699	−425412	−10868	−983	−9315	−2252	−316	−459	−13	
2	−217640	−42704	−749372	−18554	−1604	−14180	−3050	−340	−256	8	
3	−279101	−52959	−887137	−20525	−1595	−11671	−1633	11	457	−1	
4	−290076	−51684	−787382	−15546	−876	−1916	1374	402	−100		
5	−244311	−38013	−448909	−4329	351	10379	3267	−129			
6	−144616	−13632	62614	9818	1508	14404	1430				
7	−5131	16583	598094	20342	1547	−8081					
8	146408	43928	908004	16796	−1144						
9	266403	55233	625974	−15810							
10	293128	32224	−748068								
11	144463	−49115									
12	−284372										
归一化常数	57228600	7498920	84987760	1307504	67184	335920	38896	2288	1144	8	

第五节　分析信号的变换方法

一、卷积运算的物理意义

变换方法是分析信号预处理中一种很有用的方法。很多变换方法就是直接内置于分析仪器之内，由仪器量测所得数据就已是通过这样变换预处理的。本节将要讨论的卷积运算、光谱多重效应的 Hadamard 变换就是这样的例子。此外，对于在光谱特别是红外光谱中得到极其广泛使用的傅里叶变换将在本节给出较详细的介绍，而对于近年来在近代数学得以迅速发展的小波变换及其在分析化学中的应用，则是本节介绍的重点。

卷积是统计学中一个十分重要的概念，在分析信号处理中有着十分重要的意义，为便于理解，试以一光谱分析的实例加以说明[13]。

设有一光谱线，以 $f(x)$ 表述，其真实图形如图 5-5 所示，用光谱仪器记录这一谱线，即

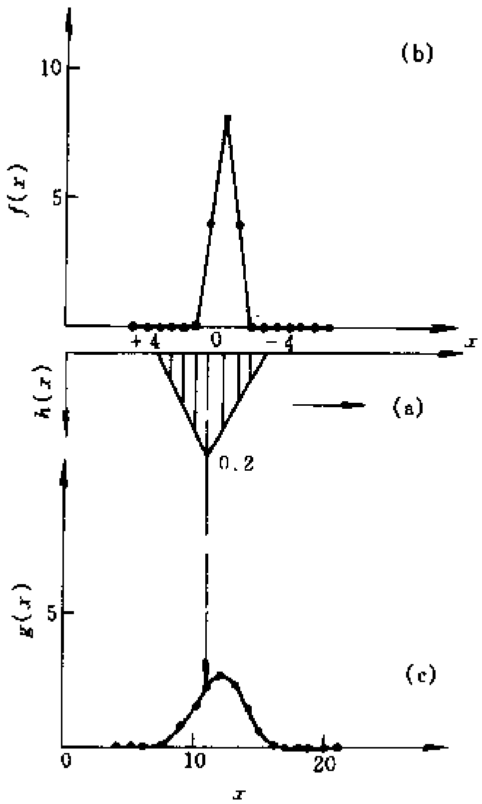


图 5-5　光谱狭缝对谱图量测的影响示意图[13]
(a) 狭缝函数 $h(x)$；
(b) 原始光谱线 $f(x)$；
(c) 光检测器测得的光谱 $g(x)$

通过一狭缝沿 x 轴进行扫描，如果狭缝是无限窄的，接受器记录的光谱应与 (b) 完全相同，但实际的光谱仪器的狭缝均有一定宽度。设这一狭缝可用函数 $h(x)$ 表述，如图 5-5 中 (a) 所示。从此图可以看出，狭缝函数（三角形函数）表述的意义，是在不同 x 值下狭缝接受的光强达到狭缝中心点所对应的光检测器的分数，也就是说，凡经狭缝透过的光均有一部分会达到中心点所对应的光检测器，函数 h 即是达到中心点光强的量度。作为示意图，图 5-5 给出的是狭缝中心点位于 $x=11$时的情况。

从图 5-5 可以看出，这种光狭缝移动检测光谱而使光谱变宽的情况，与前面讨论的窗口移动平滑法是相似的，h (x) 函数就相当于权函数，故 Savitzky 和 Golay 前述的窗口移动多项式拟合法就称为卷积法。如设 g (x) 为光检测器测得的光谱（参见图 5-5），那么，由于狭缝函数的作用而使其相当于原始的光谱 f (x) 变宽了，此时，量测到的谱线 g (x) 为

$$g[x(i)] = \sum_{i=-m}^{m} f(x) \cdot h[x(i) - x] \quad (5\text{-}31)$$

式(5-31)即为卷积运算的离散表达式，与前述窗口移动加权平均是类似的。同时，从式(5-31)可以看出，$N=2m+1$ 为狭缝宽度，式(5-31)表示的是以 $x(i)$ 为中心点时的量测光谱之强度。注意到狭缝函数 h (x)，在狭缝宽度之外皆为零，所以式(5-31)如用连续形式表出则变为

$$g[x(i)] = \int_{\infty}^{-\infty} f(x)h(x(i) - x)\mathrm{d}x \quad (5\text{-}32)$$

如记 $x(i)$为 y，则有

$$g(y) = \int_{-\infty}^{+\infty} f(x)h(y - x)\mathrm{d}x \quad (5\text{-}33)$$

式中，$g(y)$称为函数 $f(x)$与 $h(y)$的卷积，记为 $f(x) \cdot h(y)$。卷积亦称褶积，取 h 函数相对坐标折迭之意。

在光谱宽度的例子中，$f(x)$是未受到挠动的谱线(原始谱线)，$h(y-x)$是变宽函数，$g(y)$是量测到的谱线。$h(y-x)$这个变宽函数不一定局限于上例所述狭缝有限宽度引起的变宽。任何一种光谱变宽都可以用卷积积分来表述，如原于光谱线的多普勒变宽、碰撞变宽等。以后我们还可以看到，利用傅里叶变换可很容易地进行卷积的重要运算，卷积运算实质上也是一种变换运算。

二、光谱多重性效益与 Hadamard 变换

光谱分析的多重性效益的基本思路可以称量试验设计来加以说明[14]。设有 4 个未知重量合金样，按传统方法称量每一个试样，设称量得到方差为 σ^2，则单独称量 $x_i(i=1,2,3,4)$得到的结果的方差为 σ^2。但如设计一试验，同时将此 4 个合金样以不同组合置于双天平两侧，如：

$$
\begin{cases}
m_1 = x_1 + x_2 + x_3 + x_4 + e_1 & \text{(5-34a)} \\
m_2 = x_1 - x_2 - x_3 + x_4 + e_2 & \text{(5-34b)} \\
m_3 = x_1 - x_2 + x_3 - x_4 + e_3 & \text{(5-34c)} \\
m_4 = x_1 + x_2 - x_3 - x_4 + e_4 & \text{(5-34d)}
\end{cases}
$$

即将 4 个合金样本全置于左盘加法码 m_1；x_1、x_4 置于左盘，x_2、x_3 置于右盘加法码 m_2；余类推。由上述方程组可求出 x_i，例如 x_1 的估计值

$$\hat{x}_1 = 1/4(m_1 + m_2 + m_3 + m_4) = x_1 + 1/4\sum e_i$$

可见，此时误差为原误差的 1/4，即方差为 $\sigma^2/16$。

前述的单一称量［式（5-34）］可表述为（略误差项）

$$
\begin{bmatrix} \boldsymbol{m}_1 \\ \boldsymbol{m}_2 \\ \boldsymbol{m}_3 \\ \boldsymbol{m}_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{5-35}
$$

即

$$\boldsymbol{m} = \boldsymbol{A}\boldsymbol{x} \tag{5-36}$$

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{m} = \boldsymbol{A}\boldsymbol{m} \tag{5-37}$$

在此 $A^{-1}=A$。这种计算固然十分简单，但每一样本只称量了一次，而前述经试验设计方案，同样略去误差项，可写为

$$
\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{5-38}
$$

式（5-38）右侧的矩阵称为 Hadamard 矩阵，记为 $\boldsymbol{H}$，则有

$$\boldsymbol{m} = \boldsymbol{H}\boldsymbol{x} \tag{5-39}$$

显然，$\boldsymbol{H}$ 的各元素 H_{mn} 的绝对值均为 1

$$|H_{mn}| = 1$$

逆矩阵的计算同样简便：

$$\boldsymbol{H}^{-1} = \frac{1}{N}\boldsymbol{H} \tag{5-40}$$

这一称量试验设计思想，被用于光谱分析中的试验设计，传统的扫描方法测试光谱，每次测一个波长的光强度，如改为同时测定多个波长的光强度，若有 N 个波长，进行 N 次不同的组合测试，应能减少测试误差，或改善光谱量测的信噪比。按上述方法设计试验，要有＋1 与－1 两种取值，如改为＋1 与 0 两种取值，则技术上较易实现。实现这种测试光谱的具体方法，是将色散系统的入光狭缝或出光狭缝，由单个波长（或最小分辨）狭缝改为打开整个测试光域，但装置障板遮盖一部分波长，让其他波长处的光通过。设需测试 7 个波长处的光强度 ψ_j（$j=1, 2, \cdots, 7$），传统的方法是将色散系统转动使 ψ_1 到 ψ_7 依次通过一很窄的狭缝，每次测一个 ψ_j。而仿照称量试验设计装置的分光计，则可同时测定若干 ψ_j。如设计的障板在 ψ_j 处任其通过，记为 $1\times\psi_j$，如在 ψ_j 处有障板使光不能通过，记为 $0\times\psi_j$。这种 0－1 障板如有下列顺序 1001011，则测得的总光强为

$$\psi = 1\times\psi_1 + 0\times\psi_2 + 0\times\psi_3 + 1\times\psi_4 + 0\times\psi_5 + 1\times\psi_6 + 1\times\psi_7$$

$$=\psi_1+\psi_4+\psi_6+\psi_7$$

只要能设计 7 个相互独立的量测组合，测出 7 个总 ψ 值、即可解出各 ψ_j。下述矩阵就是这种组合之一：

$$S=\begin{bmatrix}1&0&0&1&0&1&1\\1&1&0&0&1&0&1\\1&1&1&0&0&1&0\\0&1&1&1&0&0&1\\1&0&1&1&1&0&0\\0&1&0&1&1&1&0\\0&0&1&0&1&1&1\end{bmatrix} \tag{5-41}$$

式（5-41）这一矩阵称为 Sylvester 矩阵，系从 Halamard 矩阵导出。这个组合将第一行居最后的元素置于下行最前作第一元素，而 1～6 元素则右移一位排到第二行，余类推。一般说来，如需量测 N 个光谱单元，S 矩阵由元素 S_{ij}组成，S_{ij}取值为 0 或 1。共有 L 个组合，每个组合（$i=1$，…，L）测得的光强将为$\sum S_{ij}\psi_j$，加量测误差得

$$Y_i=\sum_{j=1}^{N}S_{ij}\psi_j+e_i \quad (i=1,\cdots,L) \tag{5-42}$$

这种量测方法可称为光谱量测的编码过程，这里用的是 Hadamard 编码，这样设计的光谱仪器称为 Hadamard 变换光谱。

借 Hadamard 编码构造光谱仪器并不需要制作 N 个这种不同组合的障板。例如 $N=3$ 时，可构造一个（11011）的障板，其最左 3 个位置相当于 110，向右移动一个位量测，得 101，再移一位，得 011。即得 $N=3$ 的 Hadamard 变换阵。这样，构造一个含 $2N-1$ 狭缝的障板，可代替 N 个独立障板。对任意 $N=2r-1$（r 为正整数）均可实现这样的狭缝系列，得到例如

$$\begin{bmatrix}y_1\\y_2\\y_3\end{bmatrix}=\begin{bmatrix}1&1&0\\1&0&1\\0&1&1\end{bmatrix}\begin{bmatrix}\psi_1\\\psi_2\\\psi_3\end{bmatrix} \tag{5-43}$$

即

$$y=S\psi \tag{5-44}$$

$$\psi=S^{-1}y \tag{5-45}$$

S^{-1}可通过将 S 中的 0 换成-1 得到，式（5-45）S^{-1}为

$$S^{-1}=\frac{1}{2}\begin{bmatrix}1&1&-1\\1&-1&1\\-1&1&1\end{bmatrix} \tag{5-46}$$

因此，解码过程是十分简便的。

Hadamard 编码使光谱分析信噪比提高 $(N+1)/2N^{1/2}$倍，在 N 很大时，约为 $N^{1/2}/2$ 倍。

这种光谱多重性效益，又称 Fellgett 效益，其原理与前面称量设计中说明的误差降低的情况相同，用 Hadamard 变换设计的红外光谱与常规红外光谱比较，确有改善信噪比的效果。

Hadamard 变换与下节讨论的傅里叶变换的不同处，在于它是基于方波型函数，而傅里叶变换则是以三角函数为基础的。除实现多重性效益外，Hadamard 变换还用于多重性分析与数据变换，例如信号压缩，化学模式识别中特征参量抽取等。一般认为傅里叶变换是更强有力的分析信号处理的手段，但 Zupan 证明[15]，Hadamard 与傅里叶变换用于红外光谱信息的压缩

效率相同，而 Hadamard 变换在运算速度上较傅里叶变换有一定优越性。与快速傅里叶变换比较，快速 Hadamard 变换由于只含加减运算，速度约为快速傅里叶变换的 8～10 倍。

三、傅里叶变换用于分析信号处理

任何随时间变化的性质均可以分解为其谐波组分，声波是最典型的例子。某种乐器的复杂声波可借多种音频振荡器组合将其重现出来。而反过来，人们聆听一个交响乐团演奏时，可分辨出单个乐器的声调和它们奏出的特定旋律，人耳在这里所作的工作实际就是本节要讨论的傅里叶变换。傅里叶变换这一强有力的数学工具的建立，本身就是人类认识自然能力的体现。19 世纪之初，J. B. J. Fourier 在其繁忙的社会活动及研究埃及文明，出版 21 卷巨著之余，在数学、物理学方面也作出了非凡的贡献。他研究热传导的分析理论，证明可用正弦与余弦组成的级数表达热传导过程，这就是熟知的数学分析中的傅里叶级数，傅里叶将这些概念推演到傅里叶积分。这些数学工具使人们习惯的时域波形可变换到频域进行考察。这样杰出的科学成就远远超出了傅里叶同时代的许多人所能理解接受的程度，直到傅里叶去世前 2 年(1828 年)，仍有当时颇具影响的数学家把某些“级数”贬为“魔鬼的创造”，认为用数学上这种异端邪说去推演自然现象是对科学的一种“羞辱”[16]。傅里叶理论的价值在他去世后一个多世纪才逐步为科学界所认识。

傅里叶级数得出的是离散频谱，是时域的波形变换中的频率信息。傅里叶级数可写成

$$x(t) = a_0 + \sum_{n=1}^{\infty} a_n\cos(nw_0t) + b_n\sin(nw_0t) \tag{5-47}$$

式中

$$w_0 = 2\pi f_0$$

$$f_0 = 1/t$$

a_0 代表了“直流项”，由 $x(t)$ 在一个周期(T)内的均值确定。

$$a_0 = \frac{1}{T}\int_0^T x(t)\mathrm{d}t \tag{5-48}$$

其余系数 a_n,b_n 为

$$a_n = \frac{2}{T}\int_0^T x(t)\cos(nw_0t)\mathrm{d}t$$

$$b_n = \frac{2}{T}\int_0^T x(t)\sin(nw_0t)\mathrm{d}t \qquad (n = 1,2,3,\cdots)$$

因此傅里叶分析是频谱分析，这与棱镜或光栅将化学激发光源发生的光分解为不同波长的谱线是完全类同的。

傅里叶级数是研究周期函数的工具，而傅里叶积分则用于研究非周期函数。为了过渡到傅里叶积分，令 $\cos(2\pi nf_0t)$ 表达为

$$\cos(2\pi nf_0t) = \frac{e^{j2\pi nf_0t} + e^{-j2\pi nf_0t}}{2} \tag{5-49a}$$

$$\sin(2\pi nf_0t) = \frac{e^{j2\pi nf_0t} - e^{-j2\pi nf_0t}}{2j} \tag{5-49b}$$

经变换，傅里叶级数可表达为

$$x(t) = \sum_{n=-\infty}^{\infty} C_n e^{j2\pi nf_0t} \tag{5-50}$$

C_n 在 n 等于 $-\infty,\cdots,-1,0,1,2,\cdots,\infty$ 时取值

$$C_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)\mathrm{e}^{-j2\pi nf_0t}\mathrm{d}t \tag{5-51}$$

对于每一个 n,C_n 将得出 $x(t)$具有频率 $f=nf_0$ 的谐波组分的振幅和相。有了这些基本概念,将直接给出傅里叶变换的定义。

设有时间函数 $x(t)$,其傅里叶变换表达式为

$$x(f) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-j2\pi f_0t}\mathrm{d}t \tag{5-52}$$

如对参量 f(频率变量)的每一个值,该积分都成立,则上式确定了 $x(t)$的傅里叶变换为$X(f)$。这里按习惯将时间函数写为 $x(t)$,其傅里叶变换写作 $X(f)$。可以设想,在光谱化学分析中从物质的电磁波解码得到光谱,实质上就是求取这个电磁波的傅里叶变换,棱镜与光栅从某种意义上说,是强有力的傅里叶变换器。这一事实本身就决定了傅里叶变换这一数学工具对化学量测的特殊意义。傅里叶变换不限于变换激发光源发生的波形为频谱,它能将各种不同的波形从时域变换到频域,例如在电化学量测中的应用。

傅里叶逆变换定义为

$$x(t) = \int_{-\infty}^{\infty} X(f)\mathrm{e}^{j2\pi f}\mathrm{d}f \tag{5-53}$$

借式(5-53)可由 $X(f)$反过来确定 $x(t)$,也就是说借上述(5-52)与(5-53)两式可在时域函数 $x(t)$与频域函数 $X(f)$之间进行任意交换,即

$$X(f)\Leftrightarrow x(t) \tag{5-54}$$

(一) 傅里叶变换的基本性质

(1) 线性加合性

$$x(t)+y(t)\Leftrightarrow X(f)+Y(f) \tag{5-55}$$

(2) 对称性

$$x(t)\Leftrightarrow x(-f) \tag{5-56}$$

(3) 时频伸缩性

$$x(at)\Leftrightarrow \frac{1}{|a|}X\left(\frac{f}{a}\right) \tag{5-57}$$

$$\left|\frac{1}{a}\right|X\left(\frac{t}{a}\right)\Leftrightarrow X(af) \tag{5-58}$$

(4) 时移性与频移性

$$x(t-t_0)\Leftrightarrow X(f)\mathrm{e}^{-j2\pi ft_0} \tag{5-59a}$$

$$x(t)\mathrm{e}^{j2\pi f_0t}\Leftrightarrow X(f-f_0) \tag{5-59b}$$

(5) 实偶奇函数变换特性

对实偶函数:$x_e(t)=x_e(-t)$,则 $x_e(t)\Leftrightarrow X_e(f)$

对实奇函数:$x_o(t)=-x_o(-t)$,则 $x_o(t)\Leftrightarrow jX_o(f)$

(二) 离散傅里叶变换及光谱多重性效益

在化学信号预处理中,一般用到的是离散傅里叶变换,是化学量测如数字化光谱或电化学量测中常见的情况。如将 $x(t)$的各离散值在均匀时间间隔 Δ 进行记录,可得序列样本值:

$$x_n=x(n\Delta) \qquad n=\cdots,-3,-2,-1,0,1,2,3,\cdots$$

时间区间的倒数 $1/\Delta$ 称为采样速率,如 Δ 以秒计,则 $1/\Delta$ 是每秒记录的样本值个数。采样定理[17]还定义一特征的频率 f_0,称 Nyquist 临界频率:

$$f_0=1/2\Delta$$

例如，对一正弦波，临界采样频率是每周期在两点采样。根据采样定理，作为时间函数的$x(t)$是分析信号，设其频率的最高值限于 f_0 (Hz)，要确定在时间 T 内的状态，毋需知道全部瞬时值，而仅需知道该分析信号函数在一些离散时刻的样值，这些样值一经确定，该分析信号波形就可完全由这些采样值再现出来。通过在量测的周期内均匀等距采集 $2f_0T$ 个样，即可取得这一分析信号所包括的全部信息，这对设计数字化的分析仪器具有重要意义。在此仅就有限个采样点读取的时域函数的离散傅里叶变换加以阐述。

设：采集 N 个序贯样本值

$$x_K=x(t_K)\quad t_k=k\Delta\quad k=0,1,2,\cdots,N-1$$

式中，Δ 是采样区间，为简化讨论，设 N 是偶数。

从 $x(t)$到 $X(f)$，在区间 $-f_0$ 至 f_0 内不谋求得到所有 f 下的 $X(f)$值，而只要求估计其中部分离散 f_n 下的 $X(f)$：

$$f_n=n/N\Delta\quad n=-2/N,\cdots,N/2 \tag{5-60}$$

式中，n 的最大最小对应于 Nyquist 临界频率所限定的频域上下限。现试以离散值之和近似表达的连续傅里叶变换：

$$X(f_n)=\int_{-\infty}^{\infty}x(t)\mathrm{e}^{-j2\pi f_n t}\mathrm{d}t=\Delta\sum_{k=0}^{N-1}X_K\mathrm{e}^{-j2\pi f_n t_k}=\Delta\sum_{k=0}^{N-1}X_K\mathrm{e}^{-j2\pi\left(\frac{n}{N\delta}\right)K\delta}$$

$$=\Delta\sum_{K=0}^{N-1}X_K\mathrm{e}^{-j2\pi K\left(\frac{n}{N}\right)}$$

这就是所谓离散傅里叶变换(Discrete Fourier Transform DFT)，亦即

$$X(n)=\sum_{k=0}^{N-1}x(k)\mathrm{e}^{-j2\pi k\left(\frac{n}{N}\right)} \tag{5-61}$$

同理可得其逆变换为

$$X(k)=\frac{1}{N}\sum_{n=0}^{N-1}X(n)^{j2\pi K\left(\frac{n}{N}\right)} \tag{5-62}$$

从上述式(5-61)及式(5-62)可以看出，只要对时域信号(例如用宽光域的检测器)在 N 个等距时刻进行采样，得到 N 个时域内的数据点 $x(t_k)$或 $x(k)$，而每一个这样的数据点又可借 DFT 转换为频域的振幅序列 $X(f_n)$或 $X(n)$，这样，光谱多重性效益同样可借傅里叶变换来实现，对于 Hadamard 变换，$\boldsymbol{m}=\boldsymbol{Hx}$，可以采用由傅里叶变换编码组成的矩阵来实现光谱多重性效益，即

$$\boldsymbol{m}=\boldsymbol{Fx} \tag{5-63}$$

$\boldsymbol{F}$ 的元素 F_{mn}为

$$\boldsymbol{F}_{mn}=\exp(2\pi jmn/N)=\cos(2\pi mn/N)+j\sin(2\pi mn/N) \tag{5-64}$$

式中，$j=\sqrt{-1}$，易见$|F_{mn}|=1$。令 $N=4$，将有

$$\begin{bmatrix}\boldsymbol{m}_1\\\boldsymbol{m}_2\\\boldsymbol{m}_3\\\boldsymbol{m}_4\end{bmatrix}=\begin{bmatrix}1&1&1&1\\1&j&-1&j\\1&-1&1&-1\\1&-j&-1&j\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}$$

值得提出的是，用 Hadamard 变换，每次只能量测一部分频率的信号，而 DFT 则可每次量测所有频率的信号，信噪比的改善将为 $N^{1/2}$。

（三）FFT 及其在光谱与电化学量测中的应用

前小节所述的离散傅里叶变换（DFT）适用于计算机运算。如定义 w 为一复数，

$$w=e^{-2\pi j/N} \tag{5-65}$$

则 DFT 的表达式改为

$$X(n)=\sum_{k=0}^{N-1}x(k)w^{nk} \tag{5-66}$$

按式(5-66)来计算，对每一个 $X(n)$，需作 N^2 次乘法，$(N-1)$个复数加法。如此，即使使用计算机，计算量也十分大。

快速傅里叶变换(Fast Fourier Transform，FFT)的基本思路是分解式(5-66)，以减少运算。在 N 是以 2 为底的幂次时(即 $N=2^a$，a 为正整数)，这种分解十分方便，现以 $N=2^2=4$ 为例加以说明[18,19]，即

$$X(n)=\sum_{k=0}^{4-1}x_0(k)w^{nk} \qquad n=0,1,2,3 \tag{5-67}$$

式中记 $x(k)$ 为 $x_0(k)$ 以表示在未简化计算前原始的 $x(k)$ 项，将式(5-67)展开，将有

$$\begin{cases}X(0)=x_0(0)w^0+x_0(1)w^0+x_0(2)w^0+x_0(3)w^0\\X(1)=x_0(0)w^0+x_0(1)w^1+x_0(2)w^2+x_0(3)w^3\\X(2)=x_0(0)w^0+x_0(2)w^2+x_0(2)w^4+x_0(3)w^6\\X(3)=x_0(0)w^0+x_0(1)w^3+x_0(2)w^6+x_0(3)w^9\end{cases}$$

$$\begin{bmatrix}X(0)\\X(1)\\X(2)\\X(3)\end{bmatrix}=\begin{bmatrix}w^0&w^0&w^0&w^0\\w^0&w^1&w^2&w^3\\w^0&w^2&w^4&w^6\\w^0&w^3&w^6&w^9\end{bmatrix}\begin{bmatrix}x_0(0)\\x_0(1)\\x_0(2)\\x_0(3)\end{bmatrix} \tag{5-68}$$

或

$$X(n)=W^{nk}x_0(k) \tag{5-69}$$

因 $w=e^{-2\pi j/N}$ 即 $w=e^{-2\pi j(kn/N)}$，所以，当(kn/N)为正整数 A 时

$$w^{nk}=e^{-j(2\pi A)}=e^{-j*0}=1$$

另外，当 kn 不为 4 的整数倍数时，只需对$(kn/4)$取余即可，也就说如 $kn=6$ 时

$$w^6=e^{-j2\pi 6/4}=e^{-j2\pi(4/4)}e^{-j2\pi 2/4}=e^{-(j2\pi/4)2}=w^2$$

由此，式(5-68)可进一步简化为

$$\begin{bmatrix}X(0)\\X(1)\\X(2)\\X(3)\end{bmatrix}=\begin{bmatrix}1&1&1&1\\1&w^1&w^2&w^3\\1&w^2&w^0&w^2\\1&w^3&w^2&w^1\end{bmatrix}\begin{bmatrix}x_0(0)\\x_0(1)\\x_0(2)\\x_0(3)\end{bmatrix} \tag{5-70}$$

因

$$w^1=e^{-j2\pi(1/4)}=e^{-j\pi/2}=\cos(\pi/2)-j\sin(\pi/2)=-j$$
$$w^3=e^{-j2\pi(3/4)}=e^{-j3\pi/2}=\cos(3\pi/2)-j\sin(3\pi/2)=j$$
$$w^2=e^{-j2\pi 2/4}=e^{-j\pi}=\cos(\pi)-j\sin(\pi)=-1$$

故式(5-70)变为

$$\begin{bmatrix}X(0)\\X(1)\\X(2)\\X(3)\end{bmatrix}=\begin{bmatrix}1&1&1&1\\1&-j&-1&j\\1&-1&1&-1\\1&j&-1&-j\end{bmatrix}\begin{bmatrix}x_0(0)\\x_0(1)\\x_0(2)\\x_0(3)\end{bmatrix} \tag{5-71}$$

这就是前述的用傅里叶变换来实现光谱多重性效益的例子。如果再用二进制来替代复数运算，则可使傅里叶变换将乘法由 N^2 次降到 $\frac{Na}{2}$次。在此，$N=2^a$，亦即 $a=\lg N$。如 $N=1024=2^{10}$，计算量由原先 1024^2 降到$\frac{1024\times10}{2}$，从而大大加速了傅里叶变换速度。故有快速傅里叶变换之称。

对于光谱化学来说，FFT 提供了极其强有力的计算手段。根据有关红外谱图标准的要求，标准谱图元取 $2\mathrm{cm}^{-1}$分辨率，并至少在带宽（3800～450）cm^{-1}下摄制，如用 Michelson 干涉仪摄谱，每个干涉图片谱需采集 8192 个数据点，FFT 的计算量仅为一般离散傅里叶变换（DFT）的 1/1250。就是因为 FFT，才使得 FT-IR 仪器成为事实。

在电化学仪器设计中，广泛应用计算机联机处理数据[20]，FFT 是重要工具之一。在化学量测中，相关的频域是由直流到若干 MHz，较光谱化学的频率低得多，这时在时域进行直接离散采样比光谱化学量测容易实现。Smith[21]在综述傅里叶变换用于电化学分析数据处理时，曾展望FFT 的应用将为化学工作者最熟悉的诸如对数据变换手段一样普及。现代电分析化学技术，基本上是一种张弛量测，即施加一定的扰动后，量测随时间变换的响应。这种扰动量施加在处于平衡或准平衡状态的电化学池上。施加的扰动一般是电扰动（电流，电位，电荷），但亦可能施加电磁波扰动（如光量子），观测的响应参量往往也是这些参量。施加的扰动与响应之间的关系，称为系统的转移函数。在动力学研究中，对施加的脉冲扰动得到的暂态响应，需要在 2～3 个数量级的时域内，采用 10～20 种不同种选定的扰动（例如不同起始—最终电位阶跃）。如需要得到频域的响应，常规作法是变换 *dc* 电位进行正弦 *ac* 响应扫描（记录 *ac* 极谱），这样的试验至少要用 10～15 种不同输入频率以得到 2～3 个数量级范围内的量测响应谱。这种类型的试验比较费时，所以一般只用作基础性理论研究。常规化学分析测试中，只量测单一频率（基频和第二谐波交流极谱），单一时间（脉冲极谱在汞滴最终时刻作直流极谱量测）或某种时间积分响应（直流极谱平均电流及计时库仑法），在作例行测试时亦不考虑电位-电流曲线的细节，只在某个单一电位（峰电位）进行量测，或在响应值不依赖于电位的电位区间内进行量测。用 FFT 技术可完全改变这一状态。在电化学分析中，例如微分脉冲极谱分析，可逆的扩散控制电化学响应通常最适合于用作分析测试，但在分析实践中可能出现一系列问题。例如，系列偏离扩散控制过程将影响分析结果的重现性；试液背景干扰，如表面活性物存在，电极表面氧化剂存在等，均影响电极表面状态，甚至对固体电极表面作预处理亦不能解决问题。如能在分析过程中随时监控这些动力学变化因素，发现异常即采取措施，可减少其对分析结果的影响。FFT 导纳响应谱测定即能提供这种信息，可编制程序从这种导纳谱中提取有关电极过程动力学机理的信息，并采取相应对策进行必要的补偿，或至少提出有关问题的警告信息。

进行 FFT 电化学量测只需有标准的模拟式电化学仪器，通过模数转换与微机接口。FFT 计算本身是在微机上完成的，某些新型电分析技术，如半积分与半微分伏安法，如果用 FFT 将使仪器设计及数据处理大为简化。

值得提出的是，FFT 在 MATLAB 上已有专门的指令，只需书写一条指令：

$$x=t(y)$$

即可得出结果，从而使其用于分析信号处理变得极为简单。

（四）傅里叶变换用于分析信号的平滑

由于傅里叶变换可将时域或空间域的信号转换为频域信号，所以它可用于分析信号的平

滑预处理。其用于分析信号平滑处理的基本思想十分简单，实因在化学量测中，量测噪声一般可用零均的噪声表达。而零均的噪声大多频率很高，而分析信号一般在时域内皆为低频率信号。所以，在对分析信号进行时频域的傅里叶变换后只需将高频部分切去，再利用傅里叶反变换将留下的低频部分保留，即可达到平滑去噪的功效。下以具体例子加以说明。设有原始记录 0.05μg・g $^{-1}$Ca 在 O_2-Hg 火焰中测得的 422.67nm 发射光谱［图 5-6，(a)］，图 (b) 是用傅里叶变换得到的实部，这曲线的右部是高频信号，主要来自误差的贡献。如用一截断函数图 (c)，(即低频部分为 1，高频部分为 0)，截断高频部分后得 (a)，继对 (d) 作傅里叶逆变换，即得平滑后的谱图 (e)。

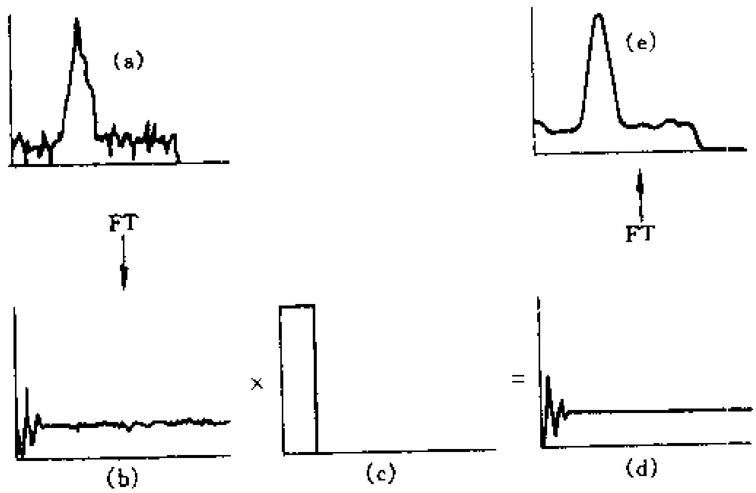


图 5-6　FT 用于发射光谱平滑处理[22]

(a) Ca 的原始发射光谱；(b) 为 (a) 的 FT 变换实部；
(c) 平滑函数；(d) 滤除噪声后的傅里叶图谱；
(e) 逆变换后还原得平滑后的光谱

当然，截断函数的选取，即正确定义高频与低频是平滑结果好坏的关键，一般可根据实际信号的情况加以选择。

(五) 傅里叶变换用于卷积与解卷积的运算

卷积与解卷积运算在电分析化学中有重要应用。Oldman[23,24]的半积分伏安法等价于求量测的电流 ϕ 与 $t^{1/2}$函数的卷积，故文献中有关方法有时称为卷积伏安法。卷积伏安法的导纳测定均是电化学数据处理中应用 FFT 成功的实例。

傅里叶变换用于卷积的解卷积运算主要基于以下的卷积定理

$$f(x)\cdot h(x)\Leftrightarrow F(U)\cdot H(U) \tag{5-72}$$

这里 $F(u)$与 $H(u)$是对应于 $f(x)$与 $h(x)$的傅里叶变换。

由式(5-72)定理，可利用下述计算步骤由原始波谱 $f(x)$与变宽函数 $h(x)$算出变宽的波谱(卷积运算)

$$f(x)\xrightarrow{\text{FT}}F(u)$$
$$h(x)\xrightarrow{\text{FT}}H(u)$$
$$H(u)F(u)-G(u)$$
$$G(u)\xrightarrow{\text{逆 FT}}g(x)$$

反之，因 $F(u)=G(u)/H(u)$，继对 $F(u)$进行傅里叶逆变换就可得 $f(x)$，即解卷积方法可由测得的(变宽的)光谱 $g(x)$与变宽函数 $h(x)$，还原出原始波谱 $f(x)$来(解卷积运算)

$$g(x)\xrightarrow{\text{FT}}G(u)$$

$$h(x)\xrightarrow{\text{FT}}H(u)$$

$$F(u)=\frac{G(u)}{H(u)}$$

$$F(u)\xrightarrow{\text{逆 FT}}f(x)$$

这种算法亦称逆滤波(inverse filtering)。这里完全忽略了误差的影响，实际量测的 $g(x)$除狭缝宽度影响外，还包括了量测误差(噪声)$n(x)$

$$g(x)=f(x)\times h(x)+n(x)$$

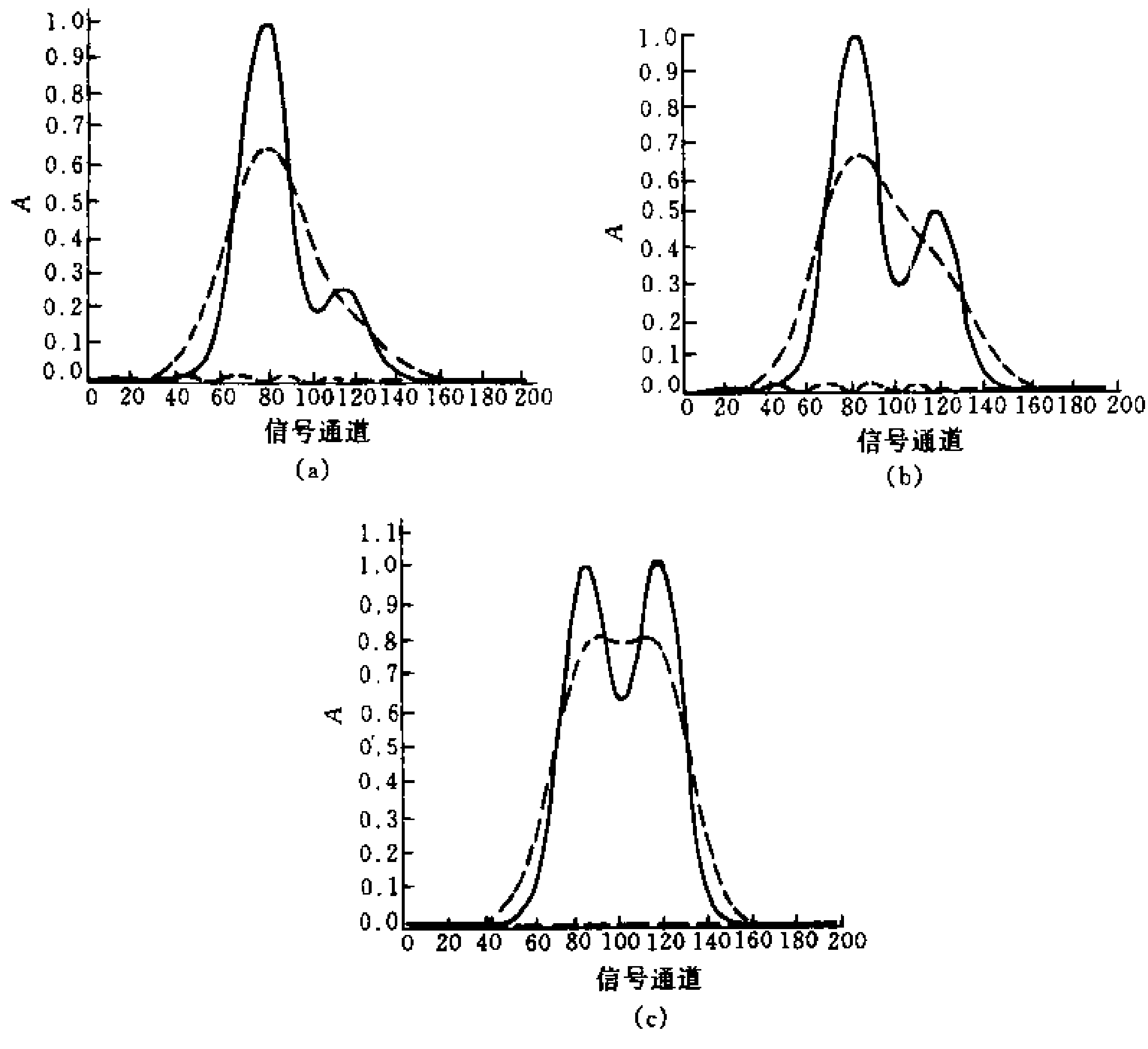


图 5-7　去卷积运算用于分辨重叠分析信号

虚线峰－测得的峰；实线峰—去卷积复原的峰；

靠近横轴的虚线—复原峰形与真实峰形之差；

h 函数与信号半宽比：(a)、(b)1.25；(c)1.0

考虑到傅里叶变换的加合性，$g(x)$的变换 $G(u)$为

$$G(u)=F(u)H(u)+N(u)$$

$$\frac{G(u)}{H(u)}=F(u)+\frac{N(u)}{H(u)}$$

或

$$\hat{F}(u)=F(u)+\frac{N(u)}{H(u)}$$

$F(u)$是未知的原始波谱的真实函数 $f(x)$的傅里叶变换，而 $\hat{F}(u)$是其估计值，实际运用卷积定理处理分析信号时，应考虑误差。

作解卷积运算的必要条件是先确定变宽函数 $h(x)$，一般可取一很窄的波谱线（至多为 $h(x)$的 1/10 宽），通过实验，确定 $h(x)$函数。此算法可用于对两个部分重叠峰的处理，还原出的原始峰形分辨情况较变宽后记录的谱图有显著改善，这是一种有用的重叠波谱分辨技术（图 5-7）。

四、小波变换用于分析信号处理

如前一小节所述，傅里叶变换的本质在于它通过三角函数将时域信号变换成频域信号，从而得到一系列有用信息，在平滑、波谱多重性效益及波谱量测方面得到了极其广泛的应用。然而，傅里叶变换也因为这一点而使其同样具有自己的局限性。正如著名数学家 Meyer 指出的那样：“当你记录了一小时信号，如果在最后五分钟出现了一些错误，可这些错误就将殃及整个傅里叶变换谱。”因为傅里叶变换旨在将整个时域信号变换为频域信号，而失去了时域中的分辨率。为此出现了窗口傅里叶变换（Windowed Fourier Transform，WET）或短时傅里叶变换（Short Time Fourier Transform，STFT）：

$$WFT_X(t,f)=\int_{-\infty}^{\infty}x(t)h(t-t')\mathrm{e}^{-j2\pi ft}\mathrm{d}t \tag{5-73}$$

其物理意义如图 5-8 所示。从图 5-8 可见，虽然傅里叶变换（谱）不能明显表示频域分量的时间局域化，但这样一种局域化可通过对信号 $x(t)$适当预加窗得到。窗口傅里叶变换可视为一局部谱。也就是说，信号 $x(t)$在时间 t'的窗口傅里叶变换就是信号 $x(t)$乘上一个以 t'为中心的“分析窗”$h(t-t')$所作的傅里叶变换。$x(t)$称为基信号，由于乘上一个相当短的窗口 $h(t-t')$，等价于取出信号在分析点 $t=t'$附近的一个切片。所以，窗口傅里叶变换直接是信号 $x(t)$在“分析时间”t'附近的“局部谱”。

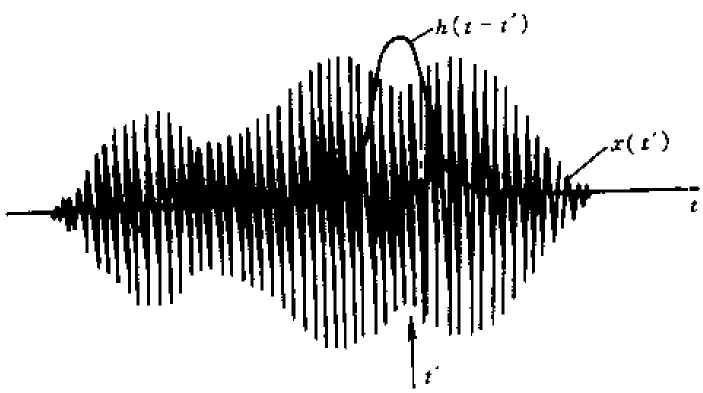


图 5-8　窗口傅里叶变换示意图

然而，尽管采用了窗口傅里叶变换可解决一些问题，但仍不是一种十分令人满意的方法。因窗口愈小时，得到时域突变信号（如尖峰或非连续信号）的能力愈强，但却对该区的低频信号无法反应；反之，窗口太大时，突变或非连续点信号就会对整个窗口的傅里叶变换谱产生影响。图 5-9 就示出了一个这样的例子。这说明，寻求一个全新的信号处理方法看来是很有必要的。

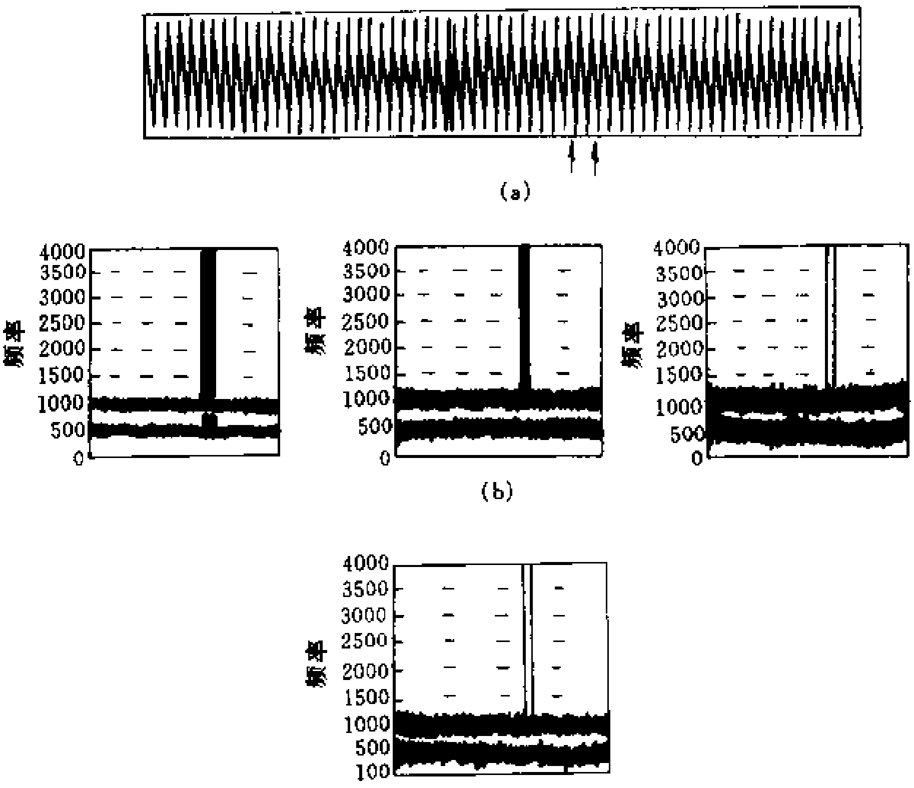


图 5-9 不同宽度的窗口傅里叶变换与小波变换结果的比较

(a)在信号接收过程中在箭头处出现两个尖锐噪声；(b)窗口 Fourier 变换的结果(左—窗口宽度太大时，在频域方向的分辨结果很好，但在时域方向却无法分辨；中—窗口宽度中等时，在频域方向的分辨结果变差，但在时域方向仍无法分辨；右—窗口宽度很小时，在频域方向的分辨结果变得很差，但时域方向已可分辨)；(c)小波变换结果(此时在频域方向的分辨结果可以接受，时域方向的分辨结果很好)

小波变换就是这样一种全新信号处理的工具，可以说是近年来在应用数学领域中独具魅力的成果。小波变换的概念最早是由法国地球物理学家 Morlet 于 20 世纪 80 年代初作为一种信号分析的数学工具而提出，很快就引起应用数学界的高度关注，并迅速在信号和图像分析、地球物理信号分析、计算机视觉与编码、语言的合成与分析、信号的奇异性检测与谱估计，甚至在分形与混沌理论中都获得广泛的应用[25]。在小波变换中，人们以不同的“标度”(scaling)和“分辨率”(resolution)同时在时域(或空间)和频域中观察信号，这种观察信号的多分辨(multiresolution)或多标度的方法是小波变换的基本点，其目的就在于“既要看到森林(信号的概貌)，又要看到树木(信号的细节)。”

小波变换最早由 Bos 等引入分析化学[26,27]。主要用于估计流动注射分析峰之强度及处理红外光谱数据。近几年来，此法在化学计量学界已引起广泛的关注，有关小波应用的文献逐年增多[28]，在我国有关小波变换在分析化学中的应用研究也得到了相当快的发展[29,34]。本小节将在介绍基本概念和算法的基础上，对小波变换在分析化学中应用的几个方面给出必要讨论。

小波变换的基本定义为

$$WT_x(a,b) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} h\left(\frac{t-b}{a}\right) x(t)\mathrm{d}t \tag{5-74}$$

从式(5-74)可以看出，它与窗口傅里叶变换[式(5-73)]很相似，不同之点只在于小波变换少了三角函数变换 ($e^{-2j\pi f t}$) 一项，但增加了一个伸缩参数 a，对于大的伸缩参数 a，基函数变成展宽的原像小波，是一个低频函数；而对于小的 a，基函数则成为缩小的小波，它是一个短

的高频函数。正是这一伸缩参数，使小波具有多分辨的（multiresolution）的性质。小波变换的另一参数为平移参数 b，这就使它也同样具有前述窗口傅里叶变换可得到时域（或空间域）"局部谱"的特性。

另外，小波变换不像傅里叶变换只采用三角函数为其基底函数，其基函数 $h(\cdot)$ 是可改变而且可以是相互正交的。不同的基函数 $h(\cdot)$ 对应于不同的小波变换，最著名和常用的小波基函数有 Haar 基，Daubechies 基和 Coiflet 基和 B-样条基等。正是因为有了这样一些特点，使小波变换成为了一种比傅里叶变化更灵活和更多样化的信号处理的数学新工具。

（一）**离散小波变换**

在分析信号的小波变换处理中，一般使用的是离散小波变换（discrete wavelet transform），如将前述小波变换定义式中的小波基函数单独列出，则有：

$$h_{a,b}(t)=(a^{-1/2})h[(t-b)/a] \tag{5-75}$$

继将其平移（translation）和伸缩或尺度（scaling）参数离散化，即

$$a=a_0^j \qquad b=ka_0^jb_0 \qquad (k,j\text{ 为正整数})$$

可得

$$WT_x(j,k)=a_0^{-j/2}\int_{-\infty}^{\infty}h(a_0^{-j}t-kb_0)x(t)\mathrm{d}t \tag{5-76}$$

特别有实用意义的是 $a_0=2$ 和 $b_0=1$ 时的小波函数，使得

$$h_{a,b}(t)=2^{-j/2}h(2^{-j}t-k)$$

式中，j,k 为任意整数。

在此 j 是可由负无穷到正无穷的整数，故式(5-76)如写成

$$h_{a,b}(t)=2^{j/2}h(2^jt-k) \tag{5-77}$$

其意义是完全没变的。

如果有一离散信号 $x(t_i)=x_i(i=1,\cdots,n)$，在不同的 j,k 下，其离散小波变换也就相当于

$$\begin{aligned}WT_x(j,k)&=\sum_{i=1}^{n}2^{-j/2}h(2^{-j}t_i-k)x_i\\&=\langle x_i,2^{-j/2}h(2^{-j}t_i-k)\rangle\end{aligned} \tag{5-78}$$

式中，$\langle\cdot,\cdot\rangle$表示内积，也就是投影，此式说明小波变换就相当于一种在小波基函数上的投影变换；不同的 j,k 代表不同的分辨率（尺度）和不同时域（平移）。而且，构成不同的小波基一般要求相互正交，即

$$\langle h_{j,k},h_{mn}\rangle=\delta_{jm}\delta_{kn}$$

$\langle\cdot,\cdot\rangle$与前同表示内积，定义为

$$\langle x,y\rangle=\int x(t)y(t)\mathrm{d}t \tag{5-79a}$$

$$\langle x,y\rangle=\sum x(t_i)y(t_i) \tag{5-79b}$$

而 δ_{jm}为 Kroneckerδ 函数，即 $\delta_{jm}=1(j=m)$或 $\delta_{jm}=0(j\neq m)$。这说明，不同的 j,k 小波基函数是正交的。如果采用不同的分辨率的时空局域，可组成一组正交基，对任意信号进行小波变换，就相当于对该信号不同分辨率的时（空）局域进行投影，可得到不同分辨率的时（空）局域的变换结果，这正是小波变换的特殊之处。图 5-10 给出了不同 j,k 下的 Daubechies 小波之例子。从图 5-10 可以看出，小波函数正是通过不同的 j,k 来调节不同的局部时域和不同分辨率的。

（二）**多分辨分析**

如前所述小波变换的最显著特性就是它的多分辨特点。用多分辨函数来逼近任一函数的

基本概念是由 Meyer 和 Mallat 提出的[35,36]，这相当于用显微镜来观察一个微生物，如果想要该微生物的总貌，则需将其调至较大视野(低频)，而不在意该微生物各部位的细节；反之，如想了解某一局部部位，则需聚焦以高分辨来观察细节。小波多分辨分析正是根据这一基本思路来进行设计的。它要求可将被解析的信号投影至一系列相互包含的空间之中，

$$\cdots \subset v_{-2} \subset v_{-1} \subset v_0 \subset v_1 \subset \cdots \subset L^2 \tag{5-80}$$

式中，L^2 表示平方可积的实数空间。

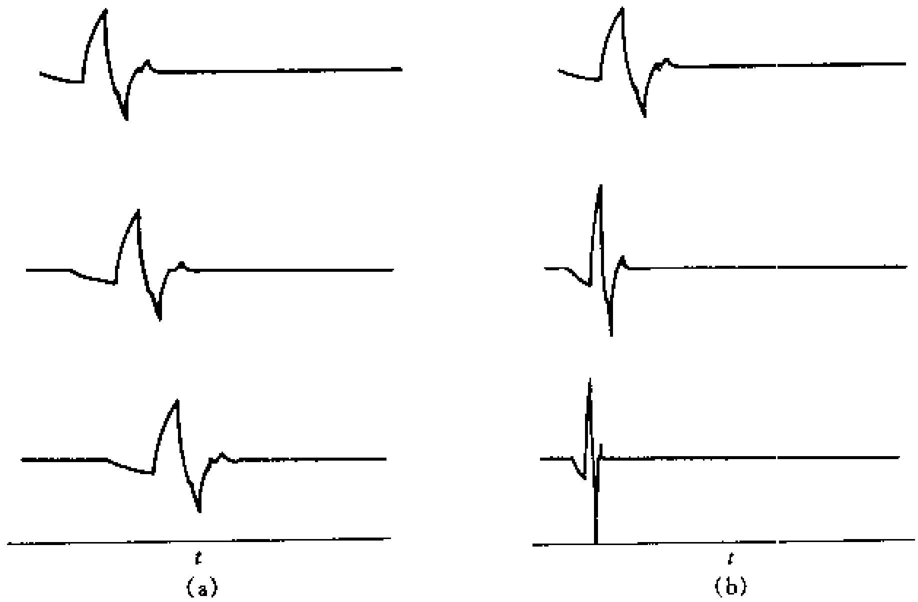


图 5-10　不同局部时域和分辨率的 Daubechies 小波

(a) 不同局部时域的 Daubechies 小波（平移）；

(b) 不同分辨率的 Daubechies 小波（尺度）

这暗示

$$v_{-\infty} = \{0\} \qquad v_{\infty} = L^2$$

由式（5-80）的上述空间的定义，上述空间自然需满足下述尺度条件

$$h(t) \in v_j \Leftrightarrow h(2t) \in v_{j+1} \tag{5-81}$$

如果仍用前节所用小波基函数的表达式 $[2^{j/2}h(2^jt-k)]$，则应有

$$h(t) = \sum_k a_k \sqrt{2}\, h(2t-k) \tag{5-82}$$

式（5-82）说明，$h(t)$（构成 v_j 的尺度基函数）就是由 $h(2t-k)$（构成 v_{j+1}的尺度基函数）的线性组合而成。在此 $h(t)$ 称为尺度函数（scaling function）或尺度滤波（scaling filter）、尺度向量（scaling vector）。a_k 则称为尺度函数系数。另外，在多分辨分析中，为表示不同尺度空间中的区别，还定义了另一类小波函数 φ_{jk}（t），并且还要求它与同尺度的尺度函数正交，即

$$< \varphi_{jk}(t), h_{jl}(t) > = \int \varphi_{jk}(t) h_{jl}(t) \mathrm{d}t = 0 \tag{5-83}$$

式中，l，j，k 都是整数。

因为 $v_0 \subset v_1 \subset v_2 \cdots \subset L$，所以，可以这样来定义小波函数的伸展空间，$w_j$（$j=0$）

$$v_1 = v_0 \oplus w_0 \tag{5-84}$$

式中，$\oplus$表示正交补（参见第十一章）根据（5-83）与（5-84）两式，还有

$$v_2 = v_0 \oplus w_0 \oplus w_1 = v_1 \oplus w_1 \tag{5-85}$$

$$L^2 = v_\infty = v_0 \oplus w_0 \oplus w_1 \oplus w_2 \oplus \cdots \tag{5-86}$$

据此，小波函数可由（5-86）式定义

$$\varphi(t)=\sum_{k} b_k \sqrt{2}\, h(2t-k) \tag{5-87}$$

式中，b_k 称为小波系数。

图 5-11 给出了尺度函数与小波矢量空间的关系。

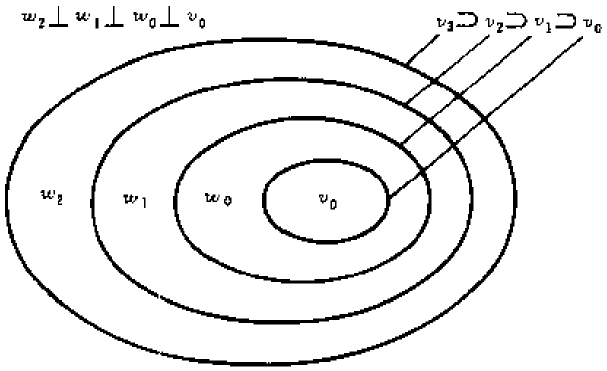


图 5-11　尺度函数与小波函数空间关系示意图

下面以 Haar 小波为例来具体说明小波多分辨分析中各类空间的具体组成[37]。

首先，Haar 小波尺度函数可定义为

$$h(t)=\begin{cases}1 & 0<t<1\\ 0 & 其他\end{cases}$$

再根据

$$h(t)=\sum_{k} a_k \sqrt{2}\, h(2t-k)$$

及

$$\varphi(t)=\sum_{k} b_k \sqrt{2}\, h(2t-k)$$

可组成不同的尺度函数与小波函数空间。在此，需指出的是，对于 Haar 小波，其尺度函数系数 α_k 只需取两个：$a_0=1/\sqrt{2}$；$a_1=1/\sqrt{2}$；而小波函数系数 b_k 也只需取两个：$b_0=1/\sqrt{2}$，$b_1=-1/\sqrt{2}$。所以，如定义上面给出的 Haar 小波尺度函数组成的是 v_0 空间的话，则 v_1 空间的尺度函数可由

$$\begin{aligned}h_1(t)&=\alpha_0\sqrt{2}\,h_0(2t)+\alpha_1\sqrt{2}\,h_0(2t-1)\\&=(1/\sqrt{2})\sqrt{2}\,h_0(2t)+(1/\sqrt{2})\sqrt{2}\,h_0(2t-1)\\&=h_0(2t)+h_0(2t-1)\\&=\begin{cases}1 & 0<t<1/2\\ 0 & 其他\end{cases}\end{aligned}$$

而小波函数 φ_0

$$\begin{aligned}\varphi_0(t)&=\frac{1}{\sqrt{2}}\sqrt{2}\,h_0(2t)-\frac{1}{\sqrt{2}}\sqrt{2}\,h_0(2t-1)\\&=h_0(2t)-h_0(2t-1)\\&=\begin{cases}1 & 0<t<1/2\\ -1 & 1/2<t<1\\ 0 & 其他\end{cases}\end{aligned}$$

由此类推，可求出 v_2，w_1 各空间的 Haar 小波的尺度基函数和小波基函数。Haar 小波基的 v_0 与 v_1 空间基函数及 w_0 与 w_1 的空间基函数图像示于图 5-12。

（三）多分辨信号分解算法

前面两小节给出了小波变换中最重要的基本概念，但实际上，小波变换的应用都是通过 Mallat 提出的多分辨信号分解（Multiresolution Signal Decomposition，MRSD）或塔式（pyramid）算法来实现的，其算法在小波变换中的地位就与快速傅里叶变换（FFT）在傅里叶变换中的地位一样重要。

如果量测所得的离散分析信号可由 $c^{(0)}=[c_1^{(0)},c_2^{(0)},\cdots,c_N^{(0)}]$来表示，其中上标(0)表示是分辨为零，即未进行任何小波变换的原始数据；N 为量测点数，为不失一般性，可假设 $N=2^p$，

（p 为正整数），则

$$c(t_i)^{(0)}=\sum c_k^{(J)}2^{J/2}h_{J,k}+\sum_{j=1}^{J}\sum_k d_k^{(j)}2^{j/2}\varphi_{j,k}$$
$$=\sum_k c_k^{(J)}2^{J/2}h(2^Jt_i-k)+\sum_j^{J}\sum_k d_k^{(j)}2^{j/2}\varphi(2^jt_i-k) \quad (5\text{-}88)$$

式中，J 表示最高分辨率；$t_i(i=1,\cdots,N)$ 表示在量测采样时间点；$(j=1,\cdots,J)$ 分别表示在 j 水平分辨率下的小波变换的尺度函数和小波函数的系数；h_{jk}，φ_{jk} 则是对应的尺度基函数和小波函数。

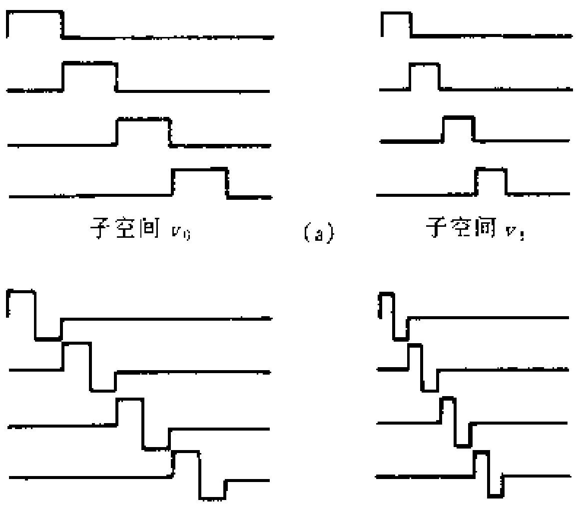


图 5-12　Haar 波空间基函数示意图

可以证明，对于不同分辨水平的尺度函数系数 $c_k^{(j)}$ 和小波函数系数 $d_k^{(j)}$，可由递归式(5-89)算出[39]：

$$c_k^{(j)}=\sqrt{2}\sum_n c_n^{(j-1)}q_{n-2k} \quad (5\text{-}89a)$$

$$d_k^{(j)}=\sqrt{2}\sum_n c_n^{(j-1)}g_{n-2k} \quad (5\text{-}89b)$$

式中，q_k 与 g_k 分别表示低通滤波器 $\boldsymbol{Q}(=\{q_k\})$ 和高通滤波器 $\boldsymbol{G}(=\{g_k\})$ 矩阵中的元素，它们应具有如下性质：

$$g_k=(-1)^k q_{1-k} \quad (5\text{-}90)$$

而且

$$\sum_k q_k=\sqrt{2}\,;\ \sum_k g_k=0$$

上述这两种系数中，q_k 是主要的，只要 q_k 确定了，g_k 也就唯一确定了。而且 q_k 的取法有多种多样，不同的小波基函数就有不同的 q_k，最简单的就要算 Haar 小波了。

$$q_0=\frac{1}{\sqrt{2}}\,;q_1=\frac{1}{\sqrt{2}} \quad (5\text{-}91a)$$

$$g_0=(-1)^0q_{1-0}=q_1=\frac{1}{\sqrt{2}} \quad (5\text{-}91b)$$

$$g_1=(-1)^1q_{1-1}=-q_0=-\frac{1}{\sqrt{2}} \quad (5\text{-}91c)$$

得到 q_k 和 g_k，利用前述递归式即可算出所有各个分辨率下的尺度函数和小波函数的系数 $c_k^{(j)}$ 和 $d_k^{(j)}$ 来，小波多分辨信号分解也就逐步进行了。所以，如何得到不同的小波变换的 q_k，是小波多分辨信号分解的关键所在。有关如何确定 q_k 的方法，读者可参阅文献[39]及文献[25]。

下面给出多分辨信号分解（MRSD）的具体矩阵算法：

（1）低通滤波矩阵 $\boldsymbol{Q}$ 与高递滤波矩阵 $\boldsymbol{G}$ 的构造

不同的小波函数（如 Haar 小波，Daubechies 小波，Coiflet 小波等）有不同的 q_k，而且同一小波函数还有不同长度 L（$-L+1$，…，0，1，…，L）。如 Dauhechies 小波就有长度为 2，4，6，8，…等多种尺度函数的系数供选择，一旦确定了长度 L 就可根据下式来构造 $\boldsymbol{Q}$ 阵和 $\boldsymbol{G}$ 阵

$$Q_{ij}=q_{j-2i} \quad (5\text{-}92a)$$

$$G_{ij}=g_{j-2i} \tag{5-92b}$$

现以 $L=2$（$D_{2*2}=D_4$），对于分析信号长度为16的 $\boldsymbol{Q}$ 阵和 G 阵的构造为例来加以具体说明。从式（5-92）可以看出 $\boldsymbol{Q}$ 阵和 $\boldsymbol{G}$ 均为（8＊16）阶矩阵，故可令

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{Q}\\ \boldsymbol{G}\end{bmatrix}_{16\times16}$$

于是

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{Q}\\ \boldsymbol{G}\end{bmatrix}=\begin{bmatrix}
q_{-1} & q_0 & q_1 & q_2 & 0 & 0 & \cdots & & & & & & \cdots & 0\\
0 & 0 & q_{-1} & q_0 & q_1 & q_2 & & & & & & & & \\
0 & 0 & 0 & 0 & q_{-1} & q_0 & q_1 & q_2 & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & q_{-1} & q_0 & q_1 & q_2 & \cdots & & \cdots & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & \vdots\\
q_1 & q_2 & 0 & 0 & \cdots & \cdots & & & & & \cdots & 0 & q_{-1} & q_0\\
g_{-1} & g_0 & g_1 & g_2 & 0 & 0 & \cdots & \cdots & & & & & \cdots & 0\\
0 & 0 & g_{-1} & g_0 & g_1 & g_2 & 0 & 0 & 0 & \cdots & & & & 0\\
0 & 0 & 0 & 0 & g_{-1} & g_0 & g_1 & g_2 & 0 & 0 & \cdots & & & 0\\
0 & 0 & 0 & 0 & 0 & 0 & g_{-1} & g_0 & g_1 & g_2 & 0 & \cdots & & 0\\
\vdots & \vdots & \vdots & \vdots & & & & & & & & & & \vdots\\
g_1 & g_2 & 0 & 0 & 0 & \cdots & \cdots & & & & & 0 & g_{-1} & g_0
\end{bmatrix} \tag{5-93}$$

在构造式（5-93）这一矩阵时，如果 $Q_{ij}=q_{j-2i}$ 得到的（$j-2i$）超过 q_k 下标 k 的范围时就都置为0，对于本例因为所取长度为2，故只有 q_{-1}，q_0，q_1，q_2 有数值。而当 $k>2$ 或 $k<-1$ 时，Q_{ij} 皆等于零。另外，在 $\boldsymbol{Q}$ 和 G 矩阵的最后一列，即 $i=8$、$j=1$，2时，$Q_{ij}=q_{j-2i}=q_{-15}$ 或 q_{-16} 都超出 q_{-1}，q_0，q_1，q_2 的范畴，应取为零，但矩阵（5-93）仍取为 q_1，q_2 或 g_1，g_2，这是为了保持矩阵 $\boldsymbol{Q}$ 和 G 的正交性，属于特例。实因

$$\boldsymbol{T}\boldsymbol{T}^t=\begin{bmatrix}\boldsymbol{Q}\\ \boldsymbol{G}\end{bmatrix}\left[\boldsymbol{Q}^t \quad \boldsymbol{G}^t\right]=\boldsymbol{I} \tag{5-94}$$

$\boldsymbol{Q}$ 及 G 矩阵的正交性可使得MRSD唯一。

（2）信号逐步分解

逐步分解以得到不同分辨率及不同长度的分解信号矢量

$$\begin{aligned}
\boldsymbol{c}_1&=\boldsymbol{Q}\boldsymbol{c}_0\\
\boldsymbol{d}_1&=\boldsymbol{G}\boldsymbol{c}_0\\
\boldsymbol{c}_2&=\boldsymbol{Q}_1\boldsymbol{c}_1=\boldsymbol{Q}_1\boldsymbol{Q}\boldsymbol{c}_0\\
\boldsymbol{d}_2&=\boldsymbol{G}_1\boldsymbol{c}_1=\boldsymbol{G}_1\boldsymbol{Q}\boldsymbol{c}_0\\
&\vdots\\
\boldsymbol{c}_j&=\boldsymbol{Q}_{j-1}\cdots\boldsymbol{Q}_1\boldsymbol{Q}\boldsymbol{c}_0\\
\boldsymbol{d}_j&=\boldsymbol{G}_{j-1}\cdots\boldsymbol{Q}_1\boldsymbol{Q}\boldsymbol{c}_0
\end{aligned}$$

这一分解过程可由图5-13形象表出。

从上述说明可以看出，每经过一次分解，原信号的分解水平增加1，而信号矢量的长度减半，所以一般要求原信号矢量的长度为 $N=2^p$，当信号矢量的长度不等于 2^p 时，或丢弃两头信息量含量相对小的量测点，或增补一些量测点来满足这一要求。另外注意到，将 $\boldsymbol{Q}$ 矩阵称为低通滤波矩阵，$\boldsymbol{G}$ 矩阵称为高通滤波矩阵，所以一般说由 $\boldsymbol{Q}$ 阵投影所得的 $\boldsymbol{c}^{(j)}$ 一般包含着原信号的

低频信息，而由 G 阵投影得到的 $d^{(j)}$ 一般包含原信号的一些高频细节。值得指出的是，Q_1 阵行列数相对于 Q 阵缩小了一半，但构成原则仍同 Q 阵。

(3) 信号的逐步复原

因为 $Q^{+}=Q^{t}$ $G^{+}=G^{t}$，所以得到的多分辨信号分解矢量可很容易还原。即

$$Q^{t}_{j-1}c_j=Q^{t}_{j-1}Q_{j-1}\cdots Q_1Qc_0$$

$$c_{j-1}=Q_{j-2}\cdots Q_1Qc_0$$

$$G^{t}_{j-1}d_j=G^{t}_{j-1}G_{j-1}\cdots Q_1Qc_0$$

$$d_{j-1}=G_{j-2}\cdots Q_1Qc_0$$

因 Q^{t} 及 G^{t} 就等于 Q^{+} 和 G^{+}，这就为分解信号的复原带来很大的方便。以后可以看到就是利用这一特性，小波变换亦具有很强的去噪功能。

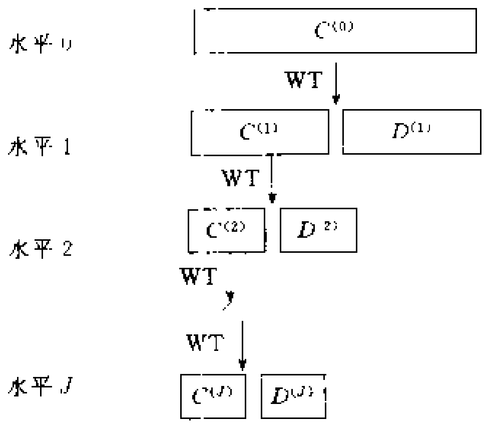


图 5-13 多分辨信号分解算法（塔式算法）示意图

（四）几个常用小波基尺度函数与小波函数系数表

如前所述，对于不同的小波基，都有不同的 q_k 和 g_k（$k=-L+1, \cdots, 0, \cdots, L$），而这些尺度函数和小波函数的系数是构成多分辨信号分解算法之关键。为方便读者，表 5-10 与表 5-11 为最常用的 3 个小波基即 Haar 小波，Daubechies 小波和 Coiflet 小波函数的系数表。

表 5-10 Daubechies 尺度及小波函数系数表

Daubechies（$L=4$）

k	q_k	g_k
−3	0.23037781330890	0.01059741078507
−2	0.71484657055292	0.03288301166689
−1	0.63088076792986	−0.03084138183556
0	−0.02798376941686	−0.18703481171909
1	−0.18703481171909	−0.02798376941686
2	−0.03084138183556	−0.63088076792986
3	0.03288301166689	−0.71484657055292
4	−0.01509740178507	−0.23037781330890

Daubechies（$L=3$）

k	q_k	g_k
−2	0.33267055295008	−0.03522629188571
−1	0.80689150931109	−0.08544127388203
0	0.45987750211849	0.13501102001025
1	−0.13501102001025	0.45987750211849
2	−0.08544127388203	−0.80689150931109
3	0.03522629188571	0.33267055295008

Daubechies（$L=2$）

k	q_k	g_k
−1	0.48292691314453	0.12940952255126
0	0.83651630373781	0.22414386804201
1	0.22414386804201	−0.83651630373781
2	−0.12940952255126	0.48292691314453

Daubechies（$L=1$）（或 Haar）

k	q_k	g_k
0	0.70710678118655	0.70710678118655
1	0.70710678118655	−0.70710678118655

表 5-11 Coiflet 尺度及小波函数系数表

Coiflet（$L=3$）

k	q_k	g_k
−2	−0.07273261951285	0.01565572813546
−1	0.33789766245781	−0.07273261951285
0	0.85257202021226	−0.38486484686420
1	0.38486484686420	0.85257202021226
2	−0.07273261951285	−0.33789766245781
3	−0.01565572813546	−0.07273261951285

Coiflet（$L=4$）

k	q_k	g_k
−3	0.04687500000000	−0.01565572813546
−2	−0.02116013576461	−0.07273261951285
−1	−0.14062500000000	−0.38486484686420
0	0.43848040729385	0.38486484686420
1	0.38486484686420	0.43848040729385
2	0.38486484686420	−0.14062500000000
3	−0.07273261951285	0.02116013576461
4	0.01565572813546	0.04687500000000

Coiflet ($L=6$)

k	q_k	g_k	k	q_k	g_k
−5	0.16387336464	0.000720549446	1	−0.076488599078	0.417005184423
−4	−0.041464936781	−0.001823208870	2	0.059434418646	0.812723635449
−3	−0.067372554722	−0.005611434819	3	0.0236680171946	0.386110066823
−2	0.386110066823	0.0236680171946	4	0.005611434819	0.067372554722
−1	0.812723635449	0.059434418646	5	−0.001823208870	−0.041464936781
0	0.417005184423	−0.076488599078	6	−0.000720549446	−0.16387336464

（五）分析信号的小波变换处理

因为小波变换实际上是一种比傅里叶变换更有力的信号处理工具，所以，近年来，在分析化学中得到了非常广泛的应用，尤其是近年来，应用面越来越广，是一个很值得引起关注的研究领域[10]，以下将主要介绍几种已取得成功应用的方法。

1. 小波变换用于数据压缩

从前节多分辨信号分解算法（MRSD）可容易看出，每经过一次变换，分析信号的长度就被压缩一半，经过几次变换，信号就可以 2 的指数速率往下降，而且，采用逐步复原的策略，只需保留 $\boldsymbol{Q}$ 阵和 $\boldsymbol{G}$ 阵，又可很容易地将分析量测信号复原，这对于有些采样点较多的一些分析信号，如红外光谱，核磁共振谱等的保存，传输以及建立较大量测波谱数据库是非常有好处的。逐步复原过程就是一个简单的矩阵乘法过程恢复速度很快，所以这一压缩技术可望在分析化学中，特别是在分析化学中波谱专家系统的构建中取到非常重要的作用。将小波变换用于紫外光谱压缩就是一个很成功的例子[40]。

另外，由于近年来联用仪器的发展相当快，往往一个样本就产生 100 万个数据点，这就为从中提取信息和进行运算带来很多困难，将小波变换的压缩功能用于这些类数据的解析之上，也是一个很成功的例子[41]，实因经小波压缩后的数据，可得主要信息，而使数据量成倍减少。

2. 小波变换用于分析信号的背景扣除

如前述小波变换的一个最大特点就是靠两个矩阵即低通滤波矩阵 $\boldsymbol{Q}$ 和高通滤波矩阵 $\boldsymbol{G}$ 对分析信号进行不同分辨率的投影运算，一般说来由低通滤波矩阵投影得到的为不同分辨率的低频信号，而高通滤波矩阵投影得到的为不同分辨率的高频信号细节，而由仪器分析所得的分析量测信号背景漂移一般可由最低频信号给出，借助分析人员的工作经验，不同分辨率所得之低频信号往往可很好地将背景与纯信号分离。文献[43]将此技术用于二维联用色谱的化学秩估计问题（见第五章）。将此法用于极谱峰的背景扣除亦是一个很好的例子[42]。

3. 小波变换用于平滑除噪

除噪（或平滑）在分析信号处理中是一个相当重要的任务，寻求一种很好的除噪方法，一直是化学计量学值得研究的问题。小波多分辨信号分解算法（MRSD）亦可用于除噪[43~45]，它一般可通过对高通滤波矩阵的投影信号（或称小波域 wavelet domain）取阈值（Threshold）的方法来进行。数学家们在小波分析的这一领域中做了不少工作[43~45]。在此，我们仅给出一例来形象说明小波变换如何利用取阈值来进行除噪的过程。

图 5-14 给出的是一个用小波来对核磁共振谱进行除噪的例子。

从图 5-14 例子可以看，小波用于除噪似乎与傅里叶变换除噪无本质区别，都是去掉变换谱中的一些值，傅里叶变换去掉的是高频部分。然而两者除噪的原理是很不相同的，值得提出的是小波变换除噪后是有可能保存高频信号的，用于高频信号除噪当优于傅里叶变换。

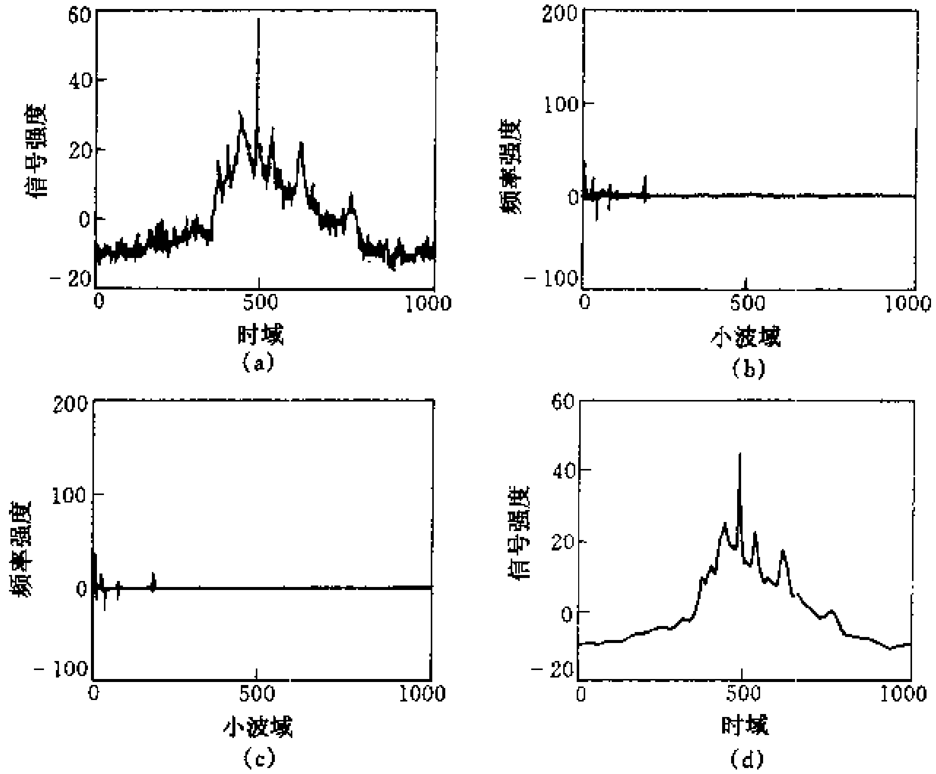


图 5-14　小波变换用于核磁共振谱除噪的一个例子

(a) 原始核磁共振谱；(b) 小波变换的细节矢量；

(c) 去掉低于阈值以下元素的细节矢量；(d) 去噪后的重构核磁共振谱

4. 小波变换在分析化学其他方面的应用

实际小波变换还可用于其他方面，如图像处理[46,47]，因小波多分辨分析可容易拓广至二维，具有很大的灵活性。另外，小波多分辨信号分解（MRSD），可同时给出信号的全貌（低通部分）和细节（高通投影部分），而且信号的压缩又是同时进行的，所以可以预料，它可在模式识别的特征抽取（参见第九章）中得到广泛而有趣的应用[43,48,49]。由于小波变换本身还在不断发展，可以预计，它将在分析信号处理中得到更加广泛而又深入的应用。

参 考 文 献

1 王铭文. 概率论与数理统计. 沈阳：辽宁人民出版社，1983

2 邓勃. 数理统计方法在分析测试中的应用. 北京：化学工业出版社，1984

3 张贤达. 现代信号处理. 北京：清华大学出版社，1995

4 俞汝勤. 化学计量学导论. 长沙：湖南教育出版社，1991

5 Long G L and Winefordner JD. Anal Chem，1983;55;712A

6 Liteanu C and Rica I. Microchimica Acta，1975; I：311

7 Liteanu C and Rica I. Statistical Theory and Methodology of Trace Analysis. Chapter 6. Chichester：Ellis Horwood Limited，1980

8 孙琢琏. 计算机在仪器分析中应用. 大连：大连理工大学出版社，1993

9 Savitzky A and Golay M J E. Anal Chem，1964；36：1627

10 Steinier J，Termonia Y，Deltour J. Anal Chem，1972；44：1906

11 Gorry P A. Anal Chem，1990；62：570

12 Green P J，Silverman B W. Nonparametric Regression and Generalized Linear Models. A Roughness Penalty Approach. Chapman & Hall，1994

13 Massart D L, Vandeginste B G M, Deming S N et al. Chemometrics: A textbook. Amsterdam: Elsevier, 1989
14 Marshall A G, Comisarow M B. Multichannel methods in Spectroscopy in Transform Techniques in chemistry, Ed. P. R. Griffiths, chapter 3. New York: Plenum Press, 1978
15 Zupan J, Bohance S, Razinger M et al. Anal Chim Acta, 1988; 210: 63
16 Ramirez R W. The FFT fundamentals and concepts. N J: Prentice Hall, Inc. 1985
17 Cooper J W. Data Handling in Fourier Transform Spectroscopy, in Transform Techniques in Chemistry, Ed. Griffiths P R. chapter 4. New York: Plenum Press, 1978
18 Brigham E O. The Fast Fourier Transform, Englewood Cliffs N J: Prentice Hall, 1974
19 Griffiths P R and De Haseth J A. Fourier Transform Infrared Spectroscopy. New York: John Wiley & Sons, 1986
20 高小霞等. 电分析化学导论. 北京: 科学出版社, 1986
21 Smith D E. Anal Chem, 1973; 45: 39
22 Horlick G. Anal Chem, 1972; 44: 943
23 Oldman K B. Anal Chem, 1973; 45: 39
24 Goto M and Oldman K B. Anal Chem, 1973; 45: 2043; 1974; 46: 1522
25 张贤达. 现代信号处理. 北京: 清华大学出版社, 1995
26 Bos M, Hoogendan E. Anal Chim Acta, 1992; 267: 73
27 Bos M, Vrielink J A M. Chemom Intell Lab Sys, 1994; 23: 115
28 Leung A K M, Chau FT. Chemom Intell Lab Sys, 1998; 13: 165
29 X. Shao, W. cai. J. Chemometrics, 1998; 12: 85
30 阎丽, 莫金垣. 科学通报, 1995; 40: 2567
31 Yu X, Mo J. The Analyst, 1996; 121: 1099
32 邹学广, 孙培艳, 蔡文生. 分析化学, 1997; 25: 671
33 Shen H L, Liang Y Z. Chemom Intell Lab Sys, 1997; 37: 261～269
34 沈海林, 梁逸曾. 高等学校化学学报, 1997; 16: 530～534
35 Mayer Y. Ondeletles et Operafeurs. Tome 1. Paris: Ondeletles Hermann, 1990
36 Mallat S. IEEE Trans, Pattern Anal, Machine Intell. 1989; 11: 674
37 Burrus C S, Gopinath R A, Guo H T. Introdution to Wavelets and Wavelet Transforms. New Jersey: Prentice Hall, 1998
38 Mallat S. Trans Amer Math Soc, 1989; 315: 69
39 Daubechies I. IEEE Trans Information Theory, 1990; 36: 961
40 Walczak B, Massart D L. Trends in Analytical Chemistry, 1997; 16: 451
41 Chau F T, Shih T M, Gao J B, et al. Applied Spectroscopy, 1996; 50: 339
42 陈洁, 仲红波, 潘忠孝等. 分析化学, 1996; 24: 1002～1006
43 Walczak B, Van den Bogaet B, Massart D L. Anal Chem, 1996; 86: 1742～1747
44 David. L. Donoho, De-noising by thresholding, IEEE Trans. On Information Theory, 1995; 41: 613
45 Donoho D L, and Johnstone I M. J R Statist Soc B, 1995; 57: 301～369
46 Beylkin G, Coifman R, Rokhlin V. Comm Pure Appl Math, 1991; 44: 141～183
47 Devore R, Jawerth B, Lucier B. IEEE Trans Inf Theory, 1992; 38: 719～746
48 Walczak B, Massart D L. Chemom Intell Lab Sys, 1997; 38: 39～50
49 Bos M, Vrielink J AM. Chemom Intell Lab Sys, 1994; 23: 115～122

第六章　多元校正与多元分辨

第一节　分析化学中的校正理论

一、黑、白、灰多组分体系分类

现代分析化学作为一门化学量测科学，有其本身的基本特征。国家自然科学基金委员会组织的学科发展战略调研报告[1]中指出，现代分析化学学科在当今变革中具有如下两个基本特点：①“从采用的手段看，分析化学是采用光、电、磁、热、声等物理现象基础上进一步采用数学、计算机科学及生物学等新成就对物质作全面的纵深分析的科学”；②“从解决的任务看，现代分析化学已发展成为获取形形色色物质尽可能多和尽可能全面的结构与成分信息，进一步认识自然改造自然的科学。”简言之，分析手段的仪器化和化学体系的复杂化已成为现代分析化学学科的两大重要特征。

首先考察一下现代分析化学的仪器化这一基本特征。近几十年来，大量新型仪器（如色谱，光谱，波谱，极谱等）相继问世，尽管其具体内容存在多种差别，可都是为分析化学家提供一个谱（多变量数据），而且这个谱包含有关分析化学样本的定性（含结构）与定量信息。用数学语言说，分析仪器提供的是矢量类型的数据。随着分析仪器的不断进步和联用技术的采用，有些仪器还可产生矩阵（或张量）类型的数据信息（如 GC-MS，HPLC-DAD，GC-IR，CE-DAD，HPLC-MS，荧光分析的激发发射光谱，多维核磁共振谱，等）。经典分析化学的校正方法仅以单点数据［标量］为基点，如光谱以最大吸收点，色谱以面积等来表征仪器数据，这样势必要丢失很多有用信息；对于产生矩阵类型数据的仪器，经典分析化学中的数据处理方法则更是无能为力了。这就向分析化学工作者提出了一个新问题，从这些矢量或矩阵形式的数据中，是否能抽取出比单点数据更多更重要的化学信息呢？在以后的讨论中即将看到，近年来得以迅速发展的化学计量学对此给出了肯定的回答。

另一方面，考察一下在分析化学家的研究目标——形形色色的分析试样的重要特征。目前，由于生物学、生命科学、药物学、环境科学等学科的迅猛发展，对分析化学亦提出了较高要求，即要求对复杂混合物体系，例如复杂有机混合物体系给出快速定性定量分析，正是这一要求，使得近年来色谱分析得到飞速发展。可以这样认为，纯品的分析在现代分析化学已不构成特殊困难，困难就在于混合物体系的直接快速分析。这样形形色色的多组分体系是否也存在某种特征，针对这些特征分析化学家是否有相应策略与之对应？

在实际分析工作中，我们所碰到的混合物体系，一般说来，不外乎以下三种情况：

① 对某些混合体系，其定性组成均已知，分析目的只在于对各种物种（或物种的不同形态）或部分物种进行定量分析，这样的试样包括已知药物片剂和某些已知有机反应的过程分析样本。因该体系的定性组成已完全清楚，我们将其称之为“白色分析体系”，在以后的讨论中，可以看到，除少数非线性和所谓“病态体系”外，分析化学计量学均可给出令人满意的解析结果。

② 对于分析试样毫无验前信息，即有关其物种数，哪几种化学物种及其浓度范围皆不清楚，分析化学的任务是首先确定其物种数，进而解析出各纯物种的谱图（可以是光谱、波谱，等）即先将其首先转化成为白色分析体系，然后进行定量分析。这类分析体系是分析化学中最难解析的一类体系，因其像个黑匣子，故我们称之为"黑色分析体系"。分析化学计量学家们已研究出了不少基于矩阵数据结构和张量数据结构的化学计量学解析方法，尤其是对于色谱联用仪器，如GC-MS、HPLC-DAD、GC-IR，CE DAD，HPLC-MS所产生的矩阵数据，已发展了一系列解析新方法，可望解决黑色分析体系的快速定性定量分析问题。

③ 除以上讨论的"白色"与"黑色"分析体系外，还有一类样本，其基本特征为：已知某些待测物种存在于待分析的样本，但是否存在别的未知干扰却不清楚，分析目的是在未知干扰的存在下，直接对感兴趣的待测物种进行定量分析。此类分析体系可以说是分析化学家碰得最多的一类体系，因其定性组成只部分已知，介乎于"白色"与"黑色"分析体系之间，故将其称之为"灰色"分析体系。对于这类分析体系，可以证明，只要可获得矩阵类型的数据，一般说来分析化学计量学方法可望获得有物理意义的唯一解。

从以上讨论可知，无论是从分析手段和研究对象来看，现代分析化学是一门有极其丰富内涵的化学量测科学，分析科学家面临的一个基础性课题，是研究怎样才能更合理有效地采用已有的多种化学量测手段，以达到解决不同类型的复杂多组分分析体系的定性定量解析问题的目的。这一基础课题的研究，构成现代分析化学基础理论研究的重要组成部分。

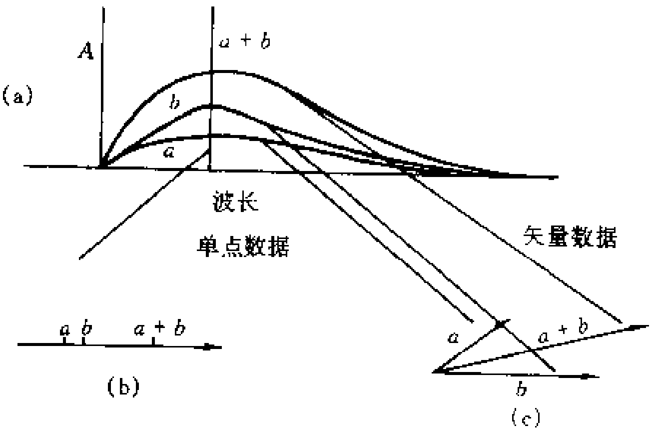


图 6-1 单点数据与矢量数据的信息差别

(b) a，b—分别表示两种化学物质，因它们都在某一波长存在吸收，如 (a) 所示，故共存时无从辨析；a，b 意义同单点数据，因不同物质具有不同波谱矢量，故在矢量空间中可表示成两个完全不同矢量，如图 (c) 所示，这为它们共存时的辨析提供了有用信息

经典分析化学的校正方法的基点是以单点数据（标量），如某一物理和化学的信号与分析体系中某一待测物质存在某种对应的数量关系（一般为线性关系），分析工作者就可借此对该化学物质进行定性定量分析。本书所论及的多元校正和多元分辨方法与上述经典校正方法在概念上已有了本质区别，它不只是用几个相应测量点来求解的传统多组分同时测定的简单推广，在这里有一个从标量校正向矢量、矩阵，甚至是张量数据解析的概念性飞跃。由于矢量（或矩阵，张量）数据比标量（单点数据）所含信息要多，由此就产生了很多新的分析方法，它们可解决很多在标量分析中被认为是不可能，甚至难以想象的问题，即在前节所讨论的所谓"灰色"与"黑色"多组分分析体系的解析问题，为分析工作者开辟了一个崭新的研究领

域。图 6-1 对矢量与单点数据的信息差别给出了简要说明，更一般的情况与多元校正和多元分辨方法的关系示于图 6-2。如果采用张量数学语言，可以把标量看成零阶张量，矢量看成一阶张量，矩阵和三维数据分别看成二阶和三阶张量，则从图 6-2 可以看出，随着分析体系的复杂程度逐渐增高，即从“白色”分析体系到“黑色”分析体系，如果想得到有物理意义的唯一解，所需数据的阶数亦应提高才行；同时也告诉我们，随着能产生多维数据的新分析仪器（新联用仪器）的出现，分析化学家们解决更复杂分析体系的能力亦愈强，从而展现了现代分析化学的广阔前景。

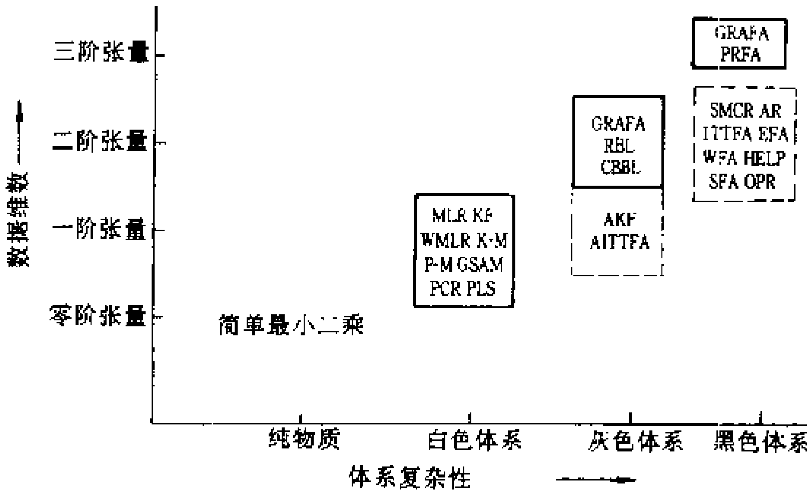


图 6-2 体系复杂性与所需仪器数据维数的关系

MLR—多元线性回归方法；KF—Kalman 滤波法；WMLR—加权最小二乘回归法；K-M—K-矩阵法；P-M—P-矩阵法；PCR—主成分回归法；PLS—偏最小二乘法；GSAM—通用标准加入法；AITTFA—标准加入迭代目标转换因子分析法；AKF—自适应 Kalman 滤波法；GRAFA—广义秩消失因子分析法；RBL—残差双线性分解法；CBBL—约束背景双线性分解法；SMCR—自模式曲线分辨法；AR—交替最小二乘回归法；ITTFA—迭代目标转换因子分析法；EFA—渐进因子分析法；WFA—窗口因子分析法；HELP—直观推导式演进特征投影法；OPR—正交投影分辨法；SFA—子窗口因子分析法；PRFA—投影旋转因子分析法

图中方框中为化学计量学近年来发展的多元校正与分辨方法，实线方框内的方法一般可得唯一解，而虚线方框内的方法除特殊情况可得唯一解，一般给出的为一定条件下的最优解。

二、张量校正理论

在此介绍的校正理论主要由 Lorber[2] 和 Kowalski 等提出[3]，这可看成为构成分析化学第二层次的基础理论的重要组成部分[4]。简言之，校正理论就是分析化学中的定量分析理论，它将常用的经典分析化学中的灵敏度、选择性、信噪比、检测下限以及结果可靠性评价等已为广大分析化学工作者广泛接受的基本概念，拓广至矢量甚至张量数据，将为分析工作者针对不同分析体系，选择正确定量校正方法，评价分析结果的可靠性提供理论依据。

表 6-1 给出了该理论提出的基本公式，在此，只对这些公式的意义加以适当说明，而有关矢量校正公式的详细推导，请参阅本章多元校正的分析化学品质因数和可靠性分析一节，还可进一步参阅文献[2,5～9]。

对于零阶张量，亦即标量，校正模型很简单，$r=cb+e$，其中 r 表示仪器响应量，c 表示待测物的浓度，e 表示量测误差；e_r 为其估计量（表 6-1）。对于这样的量测模型，一般用简单最小二乘即可求解。对此校正模型，IUPAC 给出了分析化学家常用的灵敏度、选择性、信噪比、检测下限。在此值得讨论一下的是选择性的概念。选择性一般是相对于共存干扰而言，对

一含有 n 被测物种体系，如某物种 i 的灵敏度为较大正数，其余物种的灵敏度均等于零或为接近噪声的响应，则称该分析方法对物种 i 具有完全选择性，反之，该法对物种 i 的选择性差。显然，对于标量校正模型 $r=cb+e$，一般要求最好是具有完全选择性，否则，待测物的定量就不准确了。

表 6-1　张量校正理论的基本公式

零阶张量	一阶张量		二阶张量
	直接校正	间接校正	
$r=cb+e$	$\boldsymbol{r}=\boldsymbol{Kc}+\boldsymbol{e}$	$\boldsymbol{c}=\boldsymbol{Rb}+\boldsymbol{e}$	$\boldsymbol{M}\boldsymbol{\psi}=\boldsymbol{N}\boldsymbol{\psi}\boldsymbol{\lambda}$
$c=r/b$	$\boldsymbol{c}=(\boldsymbol{K}^t\boldsymbol{K})^{-1}\boldsymbol{K}^t\boldsymbol{r}$	$\boldsymbol{b}=\boldsymbol{R}^+\boldsymbol{c}$	$NAR_N=\mathrm{rank}(M)-\mathrm{rank}(M\mid N)$
	$\boldsymbol{NAS}_i=(\boldsymbol{I}-\boldsymbol{K}_i\boldsymbol{K}_i^+)\boldsymbol{r}$	$\Vert\boldsymbol{NAS}\Vert=1/\Vert\boldsymbol{b}\Vert$	$\boldsymbol{NAS}_N=c_N\boldsymbol{x}\boldsymbol{y}^t$
	$SEL_i=\Vert\boldsymbol{NAS}_i\Vert/\Vert\boldsymbol{r}\Vert$	$SEL=\Vert\boldsymbol{NAS}\Vert/\Vert\boldsymbol{r}\Vert$	$SEL_N=\Vert\boldsymbol{NAS}_N\Vert/\Vert\boldsymbol{N}\Vert=1$
$SEN=(\Delta r/\Delta c)=b$	$SEN_i=\Vert\boldsymbol{k}_i^*\Vert$	$SEN=\Vert\boldsymbol{NAS}\Vert$	$SEN=\Vert\boldsymbol{NAS}_N\Vert$
	$\varepsilon_r=\Vert(\boldsymbol{I}-\boldsymbol{K}_i\boldsymbol{K}_i^+)\boldsymbol{r}\Vert$		$\varepsilon_r=\Vert\boldsymbol{E}\Vert$
$S/N=r/\varepsilon_r$	$S/N=\Vert\boldsymbol{NAS}_i\Vert/\varepsilon_r$		
$LOD=k\varepsilon_r/b$	$LOD_i=k\varepsilon_r/SEN_i$	$LOD=k\varepsilon_r\Vert\boldsymbol{b}\Vert$	$LOD=k\varepsilon_r/\Vert\boldsymbol{NAS}_N\Vert$

然而，对于一阶张量校正模型，情况就发生了变化。首先考察一下直接校正模型 $\boldsymbol{r}=\boldsymbol{Kc}+\boldsymbol{e}$，其中 $\boldsymbol{r}$ 表示混合物量测矢量；$\boldsymbol{K}$ 表示该混合体系所含物种的所有纯物种量测谱，也称校正矩阵；$\boldsymbol{c}$ 表示待求的混合物各物种的浓度矢量，$\boldsymbol{e}$ 为量测误差矢量。对于这样的校正模型，可容易的用多元最小二乘回归直接求出，即 $\boldsymbol{c}=(\boldsymbol{K}^t\boldsymbol{K})^{-1}\boldsymbol{K}^t\boldsymbol{r}$。这就是说，对于一阶张量校正模型，其选择性将大大提高，即在定性已知干扰存在下，仍可定量测定待求物种 i 的浓度。当然，为此得先求出该物种的所谓的纯分析信号矢量 $\boldsymbol{NAS}_i$ (net analytical signal) 来，即 $\boldsymbol{NAS}_i=(\boldsymbol{I}-\boldsymbol{K}_i\boldsymbol{K}_i^+)\boldsymbol{r}$，其中 $\boldsymbol{K}_i$ 表示删去了该混合体系待求物种的纯物种量测谱 $\boldsymbol{k}_i$ 的校正矩阵，$\boldsymbol{K}_i^+$ 表示它的广义逆（参见第十一章）。求得纯分析信号矢量 $\boldsymbol{NAS}_i$ 后，直接校正模型 $\boldsymbol{r}=\boldsymbol{Kc}+\boldsymbol{e}$ 即可化为像零阶张量模型一样，继用表 6-1 所列公式即可求得所有的分析工作者所常用的灵敏度、选择性、信噪比、检测下限来（详细推导参见本章第二节）。值得指出的是，如果各纯物质量测谱相互严重重叠或相似，这样就会导致对该物种定量分析的准确度有所下降，甚至导致该分析体系成为"病态"体系而难于准确定量（参见本章第二节），所以，其选择性还未达到完全选择。对于间接校正模型，也可求得所有的分析化学家常用的灵敏度、选择性、信噪比、检测下限，有兴趣的读者可参阅文献[8]。

如果可更进一步得到二阶张量，即矩阵形式的量测数据，例如激发发射二维荧光光谱，情况可发生更大变化。就可在未知干扰的存在下直接对待测物种进行定量解析。二阶张量的校正模型为 $\boldsymbol{M}\boldsymbol{\psi}=\boldsymbol{N}\boldsymbol{\psi}\boldsymbol{\lambda}$，在此，$\boldsymbol{M}$ 表示混合物量测矩阵数据，$\boldsymbol{N}$ 表示需待测物种的标准量测矩阵；$\boldsymbol{\psi}$ 为特征向量矩阵；$\boldsymbol{\lambda}$ 为其相对应的特征值矩阵，这在数学上是一个广义特征值问题（参见第十一章）。对于这样的校正模型，我们可首先得到待测物种的纯分析物种的秩，即 NAR_N (net analyte rank)，$NAR_N=\mathrm{rank}\ (M)\ -\mathrm{rank}\ (M\mid N)$，其中 rank $(M\mid N)$ 表示在 M 中除去 N 的秩。注意到此二阶张量校正模型的选择性 $SEL_N=\Vert\boldsymbol{NAS}_N\Vert/\Vert\boldsymbol{N}\Vert=1$，说明此类模型具有完全选择性，故可在未知干扰存在下直接对待求物种进行定量解析。所以说，随着数据维数的增加，待测物种的定量选择性也跟随着增加，这被称为维数优势。有关二阶张量模型的详细解析，读者可参阅本章第三节。

这三类校正模型的优缺点可总结列表（表 6-2）。

表 6-2 三类校正模型的优缺点

数据阶数	零阶张量校正模型	一阶张量校正模型	二阶张量校正模型
所需选择性	完全选择性	纯分析信号	纯分析物矩阵的秩
最大可分析物种数	1	分析信号通道数	分析信号通道数
干扰	不能检测，结果有偏	能检测，部分情况结果有偏	能检测，分析结果准确
信号平均	不能	$\sqrt{I}$ I 为分析通道数	$\sqrt{I\times J}$ $I\times J$ 为二维分析通道数的乘积
统计性质	简单，很清楚	复杂，已基本清楚	还需继续研究

第二节 白色分析体系的多元校正方法

白色分析体系是分析化学中常见的一类分析体系。其主要特征为：该分析体系的化学组成已知，其分析目的只是对共存的部分或全部化学组分进行定量分析。一般说来，可用下列线性加合数学模型表出

$$y=c_1x_1+c_2x_2+\cdots+c_nx_n+e \tag{6-1}$$

式中，y 表示混合物的量测矢量（或为光谱，或为其他谱）、x_i（$i=1, 2, \cdots, n$）为纯物质的量测矢量（或为纯光谱，或为其他纯物质谱），e 为量测误差矢量，一般假设为服从正态分布的等方差白噪声误差，n 为混合体系的共存物种数，c_i（$i=1, 2, \cdots, n$）为未知待估参数，一般可看成为相对浓度。如果采用矩阵表示，式（6-1）可改写为

$$y=Xc+e \tag{6-2}$$

式中，$X=[x_1, x_2, \cdots, x_n]$，可称为敏感度矩阵。$c=(c_1, c_2, \cdots, c_n)^t$ 为未知待估参数矢量，或称为待测浓度矢量。

对由式（6-1）和式（6-2）表出的多组分分析体系的数学模型，分析化学计量学根据不同的实验条件和不同仪器所产生数据的不同特性，已有不少方法进行解析，本章将以直接校正，间接校正，通用标准加入法为线索，分别加以介绍和讨论。

一、直接校正方法

直接校正方法根据所采用的不同数学方法，可分为多元线性回归方法（MLR），Kalman 滤波方法（KF）和加权最小二乘回归法（WLSR），下面将分别给予介绍。

（一）多元线性回归方法

【基本思路】

多元线性回归方法为一常规的统计解析方法，可适应于由式（6-1）或式（6-2）的未知参数的估计。此法直观简单，而且具有良好的统计特性，故在分析化学计量学中得到了广泛的应用。注意到由式（6-1）和式（6-2）表出的关系式中均含有一随机量测误差 e，为使所估量测矢量与实际量测矢量的误差最小，一般采用最小二乘方法，即使得$(\hat{y}-y)^t(\hat{y}-y)=e^te$ 趋于最小，令

$$f(c)=(\hat{y}-y)^t(\hat{y}-y)=(\hat{y}-Xc)^t(\hat{y}-Xc)=e^te=\sum e_i^2 \tag{6-3}$$

式（6-3）中，只有 c 为未知量，故可对其求导而获得 $f(c)$的最小。因为

$$\begin{aligned}f(c)&=(y-Xc)^t(y-Xc)=y^ty-y^t(Xc)-(Xc)^ty+(Xc)^t(Xc)\\&=y^ty-y^tXc-c^tXy+c^tX^tXc\\&=y^ty-2y^tXc+c^tX^tXc\end{aligned} \tag{6-4}$$

在式（6-4）的推导中，因 y^tXc 为一标量，故有 $y^tXc=c^tXy$，对式（6-4）求导可得（参见第十一

章），

$$\mathrm{d}f(\boldsymbol{c})/\mathrm{d}\boldsymbol{c}=-2\boldsymbol{X}^{\mathrm{t}}\boldsymbol{y}+2\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X}\boldsymbol{c} \tag{6-5a}$$

令式（6-5a）等于零，立即可得

$$\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X}\boldsymbol{c}=\boldsymbol{X}^{\mathrm{t}}\boldsymbol{y}$$

亦即

$$\hat{\boldsymbol{c}}=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\boldsymbol{y} \tag{6-5b}$$

式（6-5b）所求解一般称为最小二乘解。只要由式（6-2）确定的数学模型中的假设成立，即量测误差的确服从高斯正态分布且具有零均等方差的噪声，则式（6-5b）给出的解具有很多良好的统计性质。因 $\boldsymbol{X}$ 和 $\boldsymbol{y}$ 均为已知量，所以如果采用 Matlab 编程，只需一个语句可算得结果，即

$$\boldsymbol{c}=\mathrm{inv}(\boldsymbol{X}'\boldsymbol{X})*\boldsymbol{X}'*\boldsymbol{y}$$

所以，在此对多元线性回归不给出基本算法了。

【基本性质】

多元线性回归给出的解具有很多良好的统计特性，对此将给出较详细的讨论。

（1）最小二乘估计 $\hat{\boldsymbol{c}}$ 是未知浓度矢量 $\boldsymbol{c}$ 的无偏估计

记 $\boldsymbol{c}$ 的平均值为

$$\mathrm{E}(\boldsymbol{c})=\mathrm{E}\begin{bmatrix}c_1\\c_2\\\vdots\\c_n\end{bmatrix}=\begin{bmatrix}\mathrm{E}(c_1)\\\mathrm{E}(c_2)\\\vdots\\\mathrm{E}(c)\end{bmatrix}$$

于是有

$$\begin{aligned}\mathrm{E}(\hat{\boldsymbol{c}})&=\mathrm{E}[(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\boldsymbol{y}]=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\mathrm{E}(\boldsymbol{y})\\&=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\mathrm{E}(\boldsymbol{X}\boldsymbol{c}+\boldsymbol{e})\\&=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X}\boldsymbol{c}+\mathrm{E}(\boldsymbol{e})\end{aligned}$$

在直接校正模型中有关量测误差服从高斯正态分布且具有零均等方差的噪声，即

$$\mathrm{E}(\boldsymbol{e})=\boldsymbol{0}$$

和

$$\mathrm{Cov}(\boldsymbol{e})=\sigma^2\boldsymbol{I}$$

有

$$\mathrm{E}(\hat{\boldsymbol{c}})=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X}\boldsymbol{c}=\boldsymbol{c}$$

（2）最小二乘估计 $\hat{\boldsymbol{c}}$ 的方差阵为

$$\mathrm{Cov}(\hat{\boldsymbol{c}})=\sigma^2(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}$$

因为

$$\begin{aligned}\mathrm{E}(\boldsymbol{y})&=\mathrm{E}(\boldsymbol{X}\boldsymbol{c}+\boldsymbol{e})=\boldsymbol{X}\boldsymbol{c}+\mathrm{E}(\boldsymbol{e})=\boldsymbol{X}\boldsymbol{c}\\\mathrm{Cov}(\boldsymbol{y})&=\mathrm{E}\{[\boldsymbol{y}-\mathrm{E}(\boldsymbol{y})][\boldsymbol{y}-\mathrm{E}(\boldsymbol{y})]^{\mathrm{t}}\}\\&=\mathrm{E}[(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{c})(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{c})^{\mathrm{t}}]=\mathrm{E}(\boldsymbol{e}\boldsymbol{e}^{\mathrm{t}})=\sigma^2\boldsymbol{I}\end{aligned}$$

所以

$$\begin{aligned}\hat{\boldsymbol{c}}-\boldsymbol{c}&=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\boldsymbol{y}-\boldsymbol{c}=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}(\boldsymbol{X}\boldsymbol{c}+\boldsymbol{e})-\boldsymbol{c}=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\boldsymbol{e}\\\mathrm{Cov}(\boldsymbol{c})&=\mathrm{Cov}[(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\boldsymbol{y}]=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\mathrm{Cov}(\boldsymbol{y})\boldsymbol{X}(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\\&=(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{t}}\sigma^2\boldsymbol{I}\boldsymbol{X}(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}=\sigma^2(\boldsymbol{X}^{\mathrm{t}}\boldsymbol{X})^{-1}\end{aligned}$$

(3) 如果定义 $s^2=(\hat{y}-y)^t(\hat{y}-y)/(m-n)$，则有，$s^2$ 为 σ^2 的无偏估计。在这里，m 是测量点数或波长数，n 是混合体系的组分数，s^2 可称为剩余方差。

在给出证明之前，先介绍一条引理。

引理 设 $\boldsymbol{g}$ 为 n 维随机矢量，$\mathrm{E}(\boldsymbol{g})=\boldsymbol{f}$，$\mathrm{Cov}(\boldsymbol{g})=\boldsymbol{V}$，又 $\boldsymbol{A}$ 为一 n 阶常数方阵，则

$$\mathrm{E}(\boldsymbol{g}^t\boldsymbol{A}\boldsymbol{g})=\boldsymbol{f}^t\boldsymbol{A}\boldsymbol{f}+\mathrm{tr}(\boldsymbol{A}\boldsymbol{V})$$

特别地，当 $\mathrm{Cov}(\boldsymbol{g})=\sigma^2\boldsymbol{I}_{n\times n}$，有

$$\mathrm{E}(\boldsymbol{g}^t\boldsymbol{A}\boldsymbol{g})=\boldsymbol{f}^t\boldsymbol{A}\boldsymbol{f}+\sigma^2\mathrm{tr}(\boldsymbol{A})$$

证明：利用公式 $\mathrm{tr}(\boldsymbol{AB})=\mathrm{tr}(\boldsymbol{BA})$，并令

$$\boldsymbol{g}=\boldsymbol{f}+\boldsymbol{g}_0$$

式中，$\boldsymbol{f}$ 为随机矢量 $\boldsymbol{g}$ 的均值，为常量矢量，$\boldsymbol{g}_0$ 为一均值为零的随机矢量，则有

$$\mathrm{E}(\boldsymbol{g}_0\boldsymbol{g}_0^t)=\mathrm{Cov}(\boldsymbol{g}_0)=\boldsymbol{V}$$

所以

$$\begin{aligned}\mathrm{E}(\boldsymbol{g}^t\boldsymbol{A}\boldsymbol{g})&=\mathrm{E}[(\boldsymbol{f}+\boldsymbol{g}_0)^t\boldsymbol{A}(\boldsymbol{f}+\boldsymbol{g}_0)]\\&=\boldsymbol{f}^t\boldsymbol{A}\boldsymbol{f}+2\boldsymbol{f}^t\boldsymbol{A}\mathrm{E}(\boldsymbol{g}_0)+\mathrm{E}(\boldsymbol{g}_0^t\boldsymbol{A}\boldsymbol{g}_0)\\&=\boldsymbol{f}^t\boldsymbol{A}\boldsymbol{f}+\mathrm{E}[\mathrm{tr}(\boldsymbol{g}_0^t\boldsymbol{A}\boldsymbol{g}_0)]=\boldsymbol{f}^t\boldsymbol{A}\boldsymbol{f}+\mathrm{E}[\mathrm{tr}(\boldsymbol{A}\boldsymbol{g}_0\boldsymbol{g}_0^t)]\\&=\boldsymbol{f}^t\boldsymbol{A}\boldsymbol{f}+\mathrm{tr}[\boldsymbol{A}\mathrm{E}(\boldsymbol{g}_0\boldsymbol{g}_0^t)]=\boldsymbol{f}^t\boldsymbol{A}\boldsymbol{f}+\mathrm{tr}(\boldsymbol{A}\boldsymbol{V})\end{aligned}$$

在上述推导过程中，用到了求迹运算与求数学期望（或称求均值）运算可交换的原理。利用以上引理，不难证明 s^2 为 σ^2 的无偏估计。

因为

$$\begin{aligned}\mathrm{E}[s^2]&=\mathrm{E}[(\boldsymbol{y}-\hat{\boldsymbol{y}})^t(\boldsymbol{y}-\hat{\boldsymbol{y}})/(m-n)]\\&=1/(m-n)\mathrm{E}[(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{c}})^t(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{c}})]\\&=1/(m-n)\mathrm{E}[(\boldsymbol{y}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{y})^t(\boldsymbol{y}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{y})]\\&=1/(m-n)\mathrm{E}[\boldsymbol{y}^t(\boldsymbol{I}_{m\times m}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)^t(\boldsymbol{I}_{m\times m}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)\boldsymbol{y}]\end{aligned}$$

式中，$(\boldsymbol{I}_{m\times m}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)^t(\boldsymbol{I}_{m\times m}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)$ 就相当于一个常数矩阵，而

$$\mathrm{E}(\hat{\boldsymbol{y}})=\mathrm{E}(\boldsymbol{X}\hat{\boldsymbol{c}})=\boldsymbol{X}\mathrm{E}(\hat{\boldsymbol{c}})=\boldsymbol{X}\boldsymbol{c}$$

另外，$(\boldsymbol{I}_{m\times m}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)$ 为一对称等幂矩阵，故有

$$\begin{aligned}&(\boldsymbol{I}_{m\times m}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)^t(\boldsymbol{I}_{m\times m}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)\\&=\boldsymbol{I}_{m\times m}-2(\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)^t+(\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)^t(\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)^t\\&=\boldsymbol{I}_{m\times m}-(\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)\end{aligned}$$

利用前述引理，可得

$$\begin{aligned}&(\boldsymbol{X}\boldsymbol{c})^t(\boldsymbol{I}_{m\times m}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)(\boldsymbol{X}\boldsymbol{c})\\&=\boldsymbol{c}^t(\boldsymbol{X}^t\boldsymbol{X}-\boldsymbol{X}^t\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{X})\boldsymbol{c}\\&=\boldsymbol{c}^t(\boldsymbol{0})\boldsymbol{c}=0\end{aligned}\tag{6-6a}$$

$$\begin{aligned}&\mathrm{tr}(\boldsymbol{I}_{m\times m}-\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)\\&=\mathrm{tr}(\boldsymbol{I}_{m\times m})-\mathrm{tr}(\boldsymbol{X}(\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t)\\&=\mathrm{tr}(\boldsymbol{I}_{m\times m})-\mathrm{tr}((\boldsymbol{X}^t\boldsymbol{X})^{-1}\boldsymbol{X}^t\boldsymbol{X})\\&=\mathrm{tr}(\boldsymbol{I}_{m\times m})-\mathrm{tr}(\boldsymbol{I}_{n\times n})=m-n\end{aligned}\tag{6-6b}$$

结合上述式(6-6)两式成立即可得

$$\mathrm{E}[s^2]=1/(m-n)\times(0+(m-n)\sigma^2)=\sigma^2\tag{6-7}$$

(4) 只要有关量测误差为服从高斯正态分布且具有零均等方差的噪声的假设成立，最小二乘估计为相合估计，即

$$\lim \mathrm{Cov}(\hat{c}) = \lim \sigma^2 (X^t X)^{-1} = \lim \sigma^2/m[(X^t X/m)^{-1}] = \lim \sigma^2/m(F) = 0 \tag{6-8}$$

这是因为式（6-8）中的 F 矩阵为一平均协方差阵。这一关系说明，只有有关假设成立，量测点越多，估计浓度的方差越小。

（5）只要有关量测误差为服从高斯正态分布且具有零均等方差噪声的假设成立，则在全部线性无偏估计类中，最小二乘估计是唯一的一个方差一致最小估计（Gauss-Markov 定律）。因 Gauss-Markov 定律的证明需用到一些不常用的统计学定理，在此从略。

从以上讨论可以看出，最小二乘估计的确具有很好的统计学特性。故虽此方法很古老，还仍在分析化学计量学中得到广泛地应用[10~12]。因其是直接用纯物质谱来进行混合体系的浓度校正，故有直接校正之称。

在分析化学计量学的多元校正方法中，最优量测点（在光谱分析中为波长）的选择一直是一个研究课题。在分析化学中，为测定多组分，亦常用多波长方法。现在根据上述讨论的最小二乘估计的性质，还对此问题进行一下分析。

在最优量测点的选择标准中，最著名的有以下 3 种：① 估计值的均方差标准[13,14]，$MSE = \sigma^2 \mathrm{tr}\ (X^t X)^{-1}$；② 敏感度矩阵行列式标准[15,16]，$S = \det(X^t X)$；③ 敏感度矩阵条件数标准[17]，$\mathrm{Cond}(X) = \mathrm{Cond}\ (X^t X)$，在此 Cond（·）表示条件数。这 3 个标准有一定内在联系，只需进行很简单代数运算，可得

$$MSE = \sigma^2 \mathrm{tr}(X^t X)^{-1} = \sigma^2 \mathrm{tr}(Q^t X^t X Q)^{-1} = \sigma^2 \sum 1/\lambda_i \tag{6-9}$$

式中，Q 为一正交矩阵，λ_i 为（$X^t X$）经正交变换所得对角矩阵的第 i 个对角元素，常称为特征值（参阅第十一章）。

$$\begin{aligned} S &= \det(X^t X) = \det(Q^t Q)\det(X^t X) = \det(Q^t)\det(Q)\det(X^t X) \\ &= \det(Q^t)\det(X^t X)\det(Q) = \det(Q^t X^t X Q) = \prod \lambda_i \end{aligned} \tag{6-10}$$

在式(6-10)推导中 Q 和 λ_i 的意义同前，并利用到了行列式的一些运算性质(参阅第十一章)。

$$\mathrm{Cond}(X) = \mathrm{Cond}(X^t X) = \mathrm{Max}(\lambda_i)/\mathrm{Min}(\lambda_i) \tag{6-11}$$

式(6-11)是条件数的定义，其中 $\lambda_i(i=1,2,\cdots,n)$的意义同前。

从以上（6-9）~（6-11）三式可以看出，3 个数值标准都可用敏感度矩阵的协方差矩阵之特征值表出，说明了它们具有的某些共同特性。值得提出的是，这 3 个标准都对最小的特征值很敏感，说明了它们与敏感度矩阵的共线性有关。图 6-3 给出了这 3 个标准用于两个实际分析体系的数值变化与量测点数的关系，可以看出，估计值均方差标准和敏感度矩阵行列式标准的数值都随着量测点数的增加而减少，具有相近的特性。这与前面讨论的最小二乘估计的性质（4）是吻合的。敏感度矩阵条件数标准却不具备这一统计学特性[14]。从这些结果可以看出，只要有关量测误差确为高斯正态分布且具有零均等方差，则量测点越多，估计浓度的方差越小，也就是说，在一般情况下，只要不引进非线性量测点，应该是量测点越多越好。而且，从图 6-3 也可以看出，对于二组分和三组分分析体系，当只取 2 个和 3 个量测点来校正时，估计值均方差标准和敏感度矩阵行列式标准的数值都较大，即估计值均方差较大，直到量测点数目大于 10 以上时，变化才趋于平稳，这说明最好能多取些量测点来进行校正。从这一角度看，经典分析化学中用多波长方法来测定多组分似很难保证估计浓度的方差最小，随着计算技术的发展，应该用多元校正来取代。图 6-4 示出的 3 个分析体系的分析结果与波长数的关系也确实支持这一结论。

根据 Gauss-Markov 定律，在全部线性无偏估计类中，最小二乘估计是唯一的一个方差一致最小估计(特性 5)，通常也把这个定理作为最小二乘估计优越性的一个重要论据。但是在此值得提出的是，在承认这一点的同时也应当注意到：① 此定理是在量测误差确为高斯正态分布且具有零均等方差的假设的前提下才成立，而在一个具体场合下，此假设是否合理是可以讨论的，并非自然成立。② 此定理中“方差最小”的结论，是在“无偏”的限制下得到的。如果不局限于无偏估计，可否找到另一些估计，在某种意义上优于最小二乘估计？这也是可以讨论的问题。事实上，在以后的讨论中，可以看到，对于所谓的“病态”分析体系，有偏估计就具有某些优越性。

（二）**Kalman 滤波法**

Kalman 滤波为信号处理中一种最优线性递推滤波方法，因其是一种递推算法，具有计算速度快且所需计算机内存少的特点，故在工程实践中，特别在航空空间技术中得到广泛应用。Kalman 滤波首先由 Rutan 等引人化学计量学[18,19]，很快在分析化学计量学的多元校正中得到应用。递推运算是 Kalman 滤波方法的一重要特点，在讨论 Kalman 滤波方法前，似有必要对递归运算给出简要介绍。

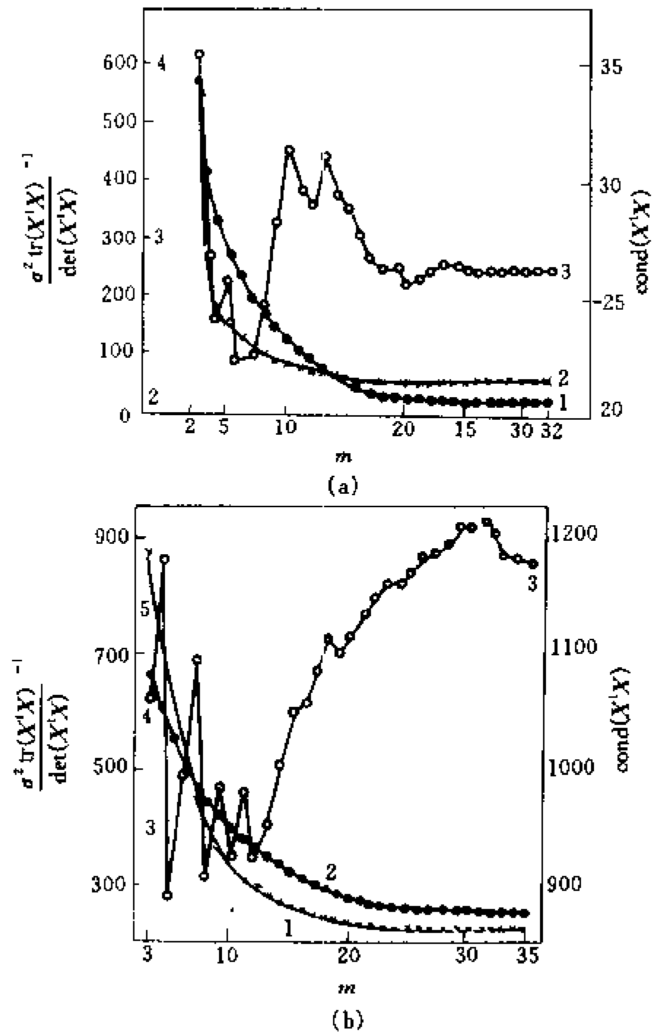


图 6-3　波长点数（m）与 3 个标准的关系

波长点序列是按最优方法选择出来的。

(a) 为苯酚与间苯二酚二组分体系；

(b) 为对-邻-间-甲酚三组分体系

1— $-\lg[\det(X'X)]$；2— $\mathrm{tr}(X'X)^{-1}$；

3— $\mathrm{cond}(X'X)$

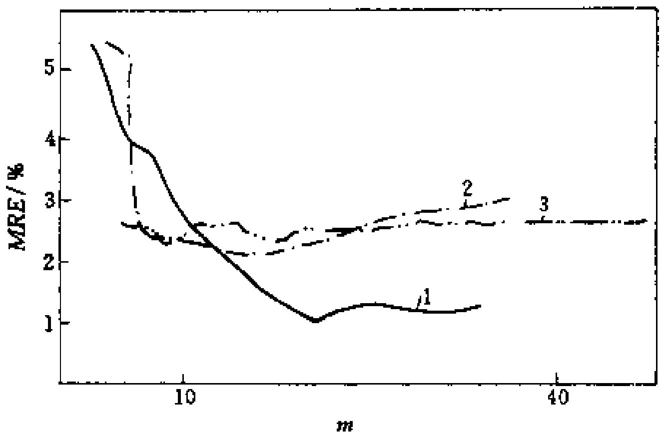


图 6-4　波长点数（m）与均方相对误差标准的关系

波长点序列是按最优方法选出

1—二组分体系；2—三组分体系；3—四组分体系

递归运算的基本思想是有效地利用已经得到的结果和最新的观察值来进行计算，以避免不必要的重复。这种递归运算的特征，可用均值的计算为例来很好地加以说明。均值运算一般采用下式

$$x=\sum x_i/n \tag{6-12}$$

式中，$\sum x_i$ 表示对 n 个观察值 x_i 求和。在测得一个新的 x_i（$i=n+1$）后，如用式（6-12）计算就必须一切从头算起，原始的 n 个观察值都必须存储在计算机中，试用递归方式运算，则可得到下列算式，它只与前一次的结果有关，而无需再一次计算原始的 n 个观察值，

$$x_n=x_{n-1}+(x_n-x_{n-1})/n \tag{6-13}$$

这种方式显然比非递归方法快且需内存少，特别适应实时运算，而这正是 Kalman 滤波方法相对于上述非递归多元线性回归方法的一重要优点。

【基本思路】

Kalman 滤波算法在化学中可用于很多方面，如化学动力学的速度常数的测定[20]，非平衡状态的电位滴定[21]，多组分化学体系的直接校正等。在 Kalman 滤波算法中，首先定义了两个模型，一为系统模型，一为量测模型。因本文只讨论在直接校正中应用的 Kalman 滤波算法，即将其看成一种递归型的线性状态估计方法，用来估计体系组分浓度，故在此只给出其简要形式，并力争与上节讨论的多元线性回归方法使用的符号一致。Kalman 滤波算法的系统模型和量测模型分别由式（6-14）与式（6-15）给出：

$$\boldsymbol{c}(k)=\boldsymbol{F}(k,k-1)\boldsymbol{c}(k-1)+\boldsymbol{w}(k) \tag{6-14}$$

$$z(k)=\boldsymbol{x}(k)^{t}\boldsymbol{c}(k-1)+e(k) \tag{6-15}$$

式中，变量 k 代表一个量测点，它可以是时间，分析通道如波长等或其他变量；$\boldsymbol{c}(k)$ 代表系统在量测点 k 的状态，在此为分析体系的浓度矢量；$\boldsymbol{F}(k,k-1)$ 称为系统的转移矩阵，它表达了系统如何由 $k-1$ 点的状态过渡到 k 点的状态，因静态分析体系的浓度矢量在整个分析过程中不发生变化，故在此它实际为一单位矩阵；$\boldsymbol{w}(k)$ 代表系统的动态随机误差，因本书只讨论静态分析体系的多组分校正，故其也可认为近似等于一零矢量；$\boldsymbol{x}(k)$ 代表量测函数矢量，它与上节讨论的多元线性回归中的敏感度矩阵 $\boldsymbol{X}$ 的第 k 行相对应；$z(k)$ 为混合体系在 k 点的量测值，与上节讨论的多元线性回归中的混合物量测矢量中的第 k 点相对应；$e(k)$ 为混合体系在 k 点的量测误差，与上节讨论的多元线性回归模型的假设相同，为一服从高斯正态分布且具有零均等方差的白噪声系列。

Kalman 滤波方法用于多元校正是采用递归来进行的，即利用系列点的量测值来进行体系浓度估计，直至得到稳定的，最优的浓度估计值。它的核心递推估计方程如下：

$$\boldsymbol{c}(k)=\boldsymbol{c}(k-1)+\boldsymbol{g}(k)[z(k)-\boldsymbol{x}(k)^{t}\boldsymbol{c}(k-1)] \tag{6-16}$$

式中，$\boldsymbol{c}(k)$表示由 k 个量测点所估计出的浓度矢量，而 $\boldsymbol{c}(k-1)$则表示由$(k-1)$个量测点所估计出的浓度矢量；$[z(k)-\boldsymbol{x}(k)^{t}\boldsymbol{c}(k-1)]$实际就表示在 k 点上混合体系的量测值与估计值之差，只不过此估计值是由 $\boldsymbol{c}(k-1)$算出，即是只基于$(k-1)$个量测点所估计出的。

式(6-16)中最重要的是 $\boldsymbol{g}(k)$，它称为 Kalman 滤波中的增益矢量，如果把式(6-16)与式(6-13)比较，可以看出此增益矢量与式(6-13)中的 $1/n$ 是相对应的。从式(6-16)还可以看到，Kalman 滤波在第 k 点对 $\boldsymbol{c}(k)$作出的估计，是建立在第$(k-1)$点以前的量测所得浓度矢量 $\boldsymbol{c}(k-1)$估计的基础上，再考虑新的第 k 点的量测值与按原估计的浓度矢量 $\boldsymbol{c}(k-1)$计算应有的响应值的差异乘上增益矢量而得。Kalman 滤波增益矢量的计算由下式给出：

$$\boldsymbol{g}(k)=\boldsymbol{P}(k-1)\boldsymbol{x}(k)[\boldsymbol{x}(k)^{t}\boldsymbol{P}(k-1)\boldsymbol{x}(k)+r(k)]^{-1} \tag{6-17}$$

式中，$r(k)$是量测噪声$e(k)$的方差，是一标量。$\boldsymbol{P}(k-1)$是从前$(k-1)$个量测点估计所得的系统协方差矩阵，其自身在第k点的估计可由式(6-18)给出：

$$\boldsymbol{P}(k)=[\boldsymbol{I}-\boldsymbol{g}(k-1)\boldsymbol{x}(k)^{t}]\boldsymbol{P}(k-1)[\boldsymbol{I}-\boldsymbol{g}(k-1)\boldsymbol{x}(k)^{t}]^{t}+\boldsymbol{g}(k-1)r(k)\boldsymbol{g}(k-1)^{t} \quad (6\text{-}18)$$

式中，$\boldsymbol{I}$为单位矩阵。

【基本算法】

从以上对 Kalman 滤波算法的介绍可以看出，只要给出$\boldsymbol{c}(k)$和$\boldsymbol{P}(k)$的初始值，即$\boldsymbol{c}(0)$和$\boldsymbol{P}(0)$，就可按式(6-17)计算其增益矢量，然后按式(6-16)和式(6-18)计算下一个$\boldsymbol{c}(k)$和$\boldsymbol{P}(k)$，直至收敛。所以，Kalman 滤波具体算法可由下列步骤组成：

(1) 置初值

$$\boldsymbol{c}(0)=\boldsymbol{0},\boldsymbol{P}(0)=\gamma^{2}\boldsymbol{I}$$

一般可取$\boldsymbol{c}(0)=\boldsymbol{0}$，$\boldsymbol{P}(0)=\gamma^{2}\boldsymbol{I}$。这里$\boldsymbol{I}$为单位矩阵，$\gamma^{2}$为初始估计的量测方差，此值可按经验式(6-19)估出：

$$\gamma^{2}=a\{r(1)/[\boldsymbol{x}(1)^{t}\boldsymbol{x}(1)]\} \quad (6\text{-}19)$$

式中，a为一与计算机精度相关的因子，一般可取 10～100。值得提出的是，$\boldsymbol{P}(0)$取值太小，可导致有偏估计，如取值太大，则尽管参量的估计值是无偏的，而由于量测点数不足够大时，$\boldsymbol{P}(0)$的最终估计值仍会偏大。

(2) 启动计算

$$\boldsymbol{g}(k)=\boldsymbol{P}(k-1)\boldsymbol{x}(k)[\boldsymbol{x}(k)^{t}\boldsymbol{P}(k-1)\boldsymbol{x}(k)+r(k)]^{-1}$$

$$\boldsymbol{c}(k)=\boldsymbol{c}(k-1)+\boldsymbol{g}(k)[z(k)-\boldsymbol{x}(k)^{t}\boldsymbol{c}(k-1)]$$

$$\boldsymbol{P}(k)=[\boldsymbol{I}-\boldsymbol{g}(k-1)\boldsymbol{x}(k)^{t}]\boldsymbol{P}(k-1)[\boldsymbol{I}-\boldsymbol{g}(k-1)\boldsymbol{x}(k)^{t}]^{t}+\boldsymbol{g}(k-1)r(k)\boldsymbol{g}(k-1)^{t}$$

式中，$r(k)$ 是量测噪声$e(k)$ 的方差，可根据实际量测噪声方差具体确定。重复此步计算直至浓度估计值趋于稳定。

【重要性质】

在 Kalman 滤波算法中，新息序列是一个很重要的概念，它可提供关于滤波结果是否可靠的信息。新息系列可由式 (6-20) 表示出：

$$v(k)=z(k)-\boldsymbol{x}(k)^{t}\boldsymbol{c}(k-1) \quad (6\text{-}20)$$

从式 (6-20) 可以看出，新息序列实际就表示在k点上的量测值与估计值之差，相当于多元线性回归中k点上的残差。如果滤波进行正常，新息序列应为一零均白噪声系列，如果滤波不正常，新息序列将成为相关的（参见图 6-5），这说明或是体系存在背景，或是体系存在未知干扰物，Kalman 滤波的结果不可靠[22]。

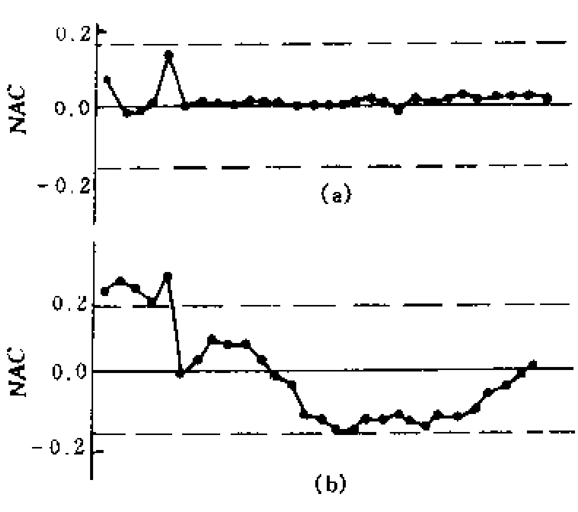


图 6-5 滤波新息序列 (NAC)
(a) 正常滤波；(b) 存在干扰物质

（三）加权最小二乘回归法

在前两节的讨论中，一般都假设量测误差是服从零均等方差的正态分布，然而，在分析化学的量测中，有时这一假设并不可能满足。在这一节中，将讨论当量测误差如是不等性方差时，多元校正是采用什么方法来进行浓度估计的。加权

最小二乘回归法就是这样一种方法。

【基本思路】

在式（6-3）中，每个误差值$e_i^2(i=1,2,\cdots,m)$的系数都是1，即在最小二乘方法的目标函数中每个误差值都是“等权”的。然而，这就暗示此法只有在各量测误差都相同的情况下才是合理的。事实上，在分析化学的量测中，量测值在不同的量测范围内，量测精度往往不同，因而量测误差也不尽相同。这种情况在光谱分析中很常见。合理的方法是对不同的误差项e_i^2加不同的权，即把式（6-3）写成

$$f(\boldsymbol{c})=\sum w_i e_i^2 \tag{6-21}$$

当量测值的精度高时，w_i大，反之，w_i小。这样可以使拟合的曲线非常接近于精度高的点，保证拟合的准确度高。

如果用矩阵的表示方法，$f(\boldsymbol{c})$可以写成

$$f(\boldsymbol{c})=\sum w_i e_i^2=\boldsymbol{e}'\boldsymbol{W}\boldsymbol{e}=(\boldsymbol{y}-\boldsymbol{y})'\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{y})=(\boldsymbol{y}\quad \boldsymbol{X}\boldsymbol{c})'\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{c}) \tag{6-22}$$

在这里，$\boldsymbol{W}$为一$m\times m$对角矩阵，其对角元素就是$w_i(i=1,2,\cdots,m)$，称为加权矩阵。

式(6-22)中，只有$\boldsymbol{c}$为未知量，故可对其求导而获得$f(\boldsymbol{c})$的最小，因为

$$\begin{aligned}f(\boldsymbol{c})&=(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{c})'\boldsymbol{W}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{c})\\&=\boldsymbol{y}'\boldsymbol{W}\boldsymbol{y}-\boldsymbol{y}'\boldsymbol{W}(\boldsymbol{X}\boldsymbol{c})-(\boldsymbol{X}\boldsymbol{c})'\boldsymbol{W}\boldsymbol{y}+(\boldsymbol{X}\boldsymbol{c})'\boldsymbol{W}(\boldsymbol{X}\boldsymbol{c})\\&=\boldsymbol{y}'\boldsymbol{W}\boldsymbol{y}-2\boldsymbol{y}'\boldsymbol{W}\boldsymbol{X}\boldsymbol{c}+\boldsymbol{c}'\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\boldsymbol{c}\end{aligned} \tag{6-23}$$

对式(6-23)求导可得

$$\mathrm{d}f(\boldsymbol{c})/\mathrm{d}\boldsymbol{c}=-2\boldsymbol{X}'\boldsymbol{W}\boldsymbol{y}+2\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\boldsymbol{c} \tag{6-24}$$

令式(6-24)等于零，立即可得

$$\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\boldsymbol{c}=\boldsymbol{X}'\boldsymbol{W}\boldsymbol{y} \tag{6-25}$$

亦即

$$\hat{\boldsymbol{c}}=(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}\boldsymbol{y} \tag{6-26}$$

这就是加权最小二乘的计算公式。在上述的推导中，完全没有考虑到随机变量的统计特性。为了比较加权最小二乘估计与多元线性回归之间的关系，假定

$$\mathrm{E}(\boldsymbol{e})=\boldsymbol{0}$$

和

$$\mathrm{Cov}(\boldsymbol{e})=\mathrm{E}(\boldsymbol{e}\boldsymbol{e}')=\boldsymbol{V}$$

在此没有要求量测误差($\boldsymbol{e}$)服从正态分布，而且量测误差的协方差矩阵也不是一个常数对角阵，即不相关且具等方差的量测误差。

【基本性质】

关于$\boldsymbol{W}$的选择问题在后面讨论。现在先讨论$\boldsymbol{W}$为一般矩阵时的加权最小二乘估计的一些基本性质。

(1) 加权最小二乘估计$\hat{\boldsymbol{c}}$是未知浓度矢量$\boldsymbol{c}$的无偏估计

$$\begin{aligned}\mathrm{E}(\hat{\boldsymbol{c}})&=\mathrm{E}[(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}\boldsymbol{y}]=(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}\mathrm{E}(\boldsymbol{y})\\&=(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}\mathrm{E}(\boldsymbol{X}\boldsymbol{c}+\boldsymbol{e})\\&=(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\boldsymbol{c}+\mathrm{E}(\boldsymbol{e})\end{aligned}$$

因仍有量测误差具有零均的性质，即$\mathrm{E}(\boldsymbol{e})=\boldsymbol{0}$，所以有

$$\mathrm{E}(\boldsymbol{c})=(\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}\boldsymbol{c}=\boldsymbol{c}$$

(2) 加权最小二乘估计 c 的方差阵为

$$\mathrm{Cov}(\hat{c})=\sigma^2(X'WX)^{-1}$$

因为

$$\begin{aligned}\hat{c}-c&=(X'WX)^{-1}X'Wy-c\\&=(X'WX)^{-1}X'W(Xc-e)-c\\&=(X'WX)^{-1}X'We\end{aligned}$$

$$\begin{aligned}\mathrm{Cov}(\hat{c})&=\mathrm{E}[(\hat{c}-c)(\hat{c}-c)']\\&=\mathrm{Cov}[(X'WX)^{-1}X'We]\\&=(X'WX)^{-1}X'W\mathrm{Cov}(e)WX(X'WX)^{-1}\\&=(X'WX)^{-1}X'WVWX(X'WX)^{-1}\end{aligned} \tag{6-27}$$

如果取 $W=V^{-1}$ 式(6-27)可变为

$$\begin{aligned}\mathrm{Cov}(\hat{c})&=(X'V^{-1}X)^{-1}X'V^{-1}VV^{-1}X(X'V^{-1}X)^{-1}\\&=(X'V^{-1}X)^{-1}X'V^{-1}X(X'V^{-1}X)^{-1}\\&=(X'V^{-1}X)^{-1}\end{aligned} \tag{6-28}$$

取 $W=V^{-1}$ 时，可使估计误差的方差为最小，因而 $W=V^{-1}$ 是最优的加权矩阵。利用许瓦茨不等式

$$A'A>(A'B)(B'B)^{-1}(B'A)$$

令 $V=C'C$，实因 V 为一对称矩阵；再令 $A=CWX(X'WX)^{-1}$，$B=(C')^{-1}X$，则

$$\begin{aligned}A'A&=(X'WX)^{-1}X'WC'CWX(X'WX)^{-1}\\&=(X'WX)^{-1}X'WVWX(X'WX)^{-1}\end{aligned} \tag{6-29}$$

又有

$$\begin{aligned}(A'B)(B'B)^{-1}(B'A)&=(X'WX)^{-1}X'WC'(C')^{-1}X[X'(C')^{-1}C^{-1}X]^{-1}X'[(C')^{-1}]'CWX(X'WX)^{-1}\\&=(X'WX)^{-1}X'WX[X'(C^{-1})'C^{-1}X]^{-1}X'WX(X'WX)^{-1}\\&=(X'WX)^{-1}X'WX[X'V^{-1}X]^{-1}X'WX(X'WX)^{-1}\\&=[X'V^{-1}X]^{-1}\end{aligned}$$

所以

$$(X'WX)^{-1}X'WVWX(X'WX)^{-1}>[X'V^{-1}X]^{-1} \tag{6-30}$$

这就说明了采用加权最小二乘时，加权阵 W 取为 V^{-1} 所得的估计误差的方差阵为最小，有时特别把加权阵 W 取为 V^{-1} 的加权最小二乘估计，即

$$\hat{c}=(X'V^{-1}X)^{-1}X'V^{-1}y \tag{6-31}$$

称为马尔柯夫估计。故一般在实际运算中就选择 V^{-1} 作为加权矩阵 W。

二、间接校正方法

从上述讨论可以看到，直接校正方法直观且计算简单，易于在计算机上实现。但是，这种方法不太利于实验设计，而且，对某些存在轻微非线性的分析体系，其浓度估计的可靠性会有所下降。对很多分析试样人们一般可得到它们的大致浓度变化区间，如果此时直接用一些已知浓度的混合物试样，继采用某些实验设计方法，如正交设计或因子设计，来构成某种训练集的话，所得估计效果当有所提高。这就产生了在这节要讨论的所谓间接校正方法。

间接校正方法的基本思路是用一些已知浓度的混合物经某种实验设计方法首先构成一校正矩阵 Y，然后，利用这校正矩阵来找到这些混合物量测谱所对应的各组分纯物种的量测谱

(K-矩阵法）或是利用它找到混合物量测谱于各混合物浓度的一种线性关系（P-矩阵法），这样一类方法因其不直接采用纯物种的标准量测谱来进行校正，故统称为间接校正方法。因间接校正方法是用混合物量测谱来直接估计未知混合物量测谱的各组分浓度，所以，有一定减轻非线性因素的功效。但是，在这里有必要指出，因间接校正方法仍是采用线性模型，对于非线性严重的分析体系仍不可能得到令人满意的结果。对于间接校正方法，其一般数学模型可由（6-32）两式给出：

$$\boldsymbol{Y}=\boldsymbol{KC} \quad (K\text{ 矩阵法}) \tag{6-32a}$$

$$\boldsymbol{C}=\boldsymbol{PY} \quad (P\text{-矩阵法}) \tag{6-32b}$$

式中，矩阵 $\boldsymbol{Y}$ 为 $(m\times p)$ 阶的校正矩阵，它是由 p 个混合物量测谱分别在 m 个分析通道（可为波长或别的分析通道）量测而得，它的每一列就代表一个混合物的量测谱；矩阵 $\boldsymbol{C}$ 为 $(n\times p)$ 阶的浓度矩阵，它是由 p 个混合物的 n 个组分的浓度构成，它的每一列表示一个混合物对应的组分浓度矢量；对于式 $\boldsymbol{Y}=\boldsymbol{KC}$，矩阵 $\boldsymbol{K}$ 为 $(m\times n)$ 阶的标准量测谱矩阵，它设为 n 个纯组分在 m 个分析通道量测所得，它的每一列代表一个纯物质的量测谱；在 $\boldsymbol{C}=\boldsymbol{PY}$ 中矩阵 $\boldsymbol{Y}$ 和矩阵 $\boldsymbol{C}$ 的意义同前；但矩阵 $\boldsymbol{P}$ 则为 $(n\times m)$ 阶的回归系数矩阵，一般要求 $m>p$ 和 $m>n$，实因所用分析通道数必须大于或至少等于纯组分数和用作校正的混合物量测谱数，否则无法求解。当然，这样的条件一般很容易满足。就是通过对上述（6-32）两式的解析，其中尤其是对式（6-32b）的解析，产生了著名的偏最小二乘（PLS）和主成分回归法（PCR）方法，下面将对这些间接校正方法作出较详细的讨论。

（一）K-矩阵法

【基本思路】

K-矩阵法首先由 Brown 引入分析化学计量学[23,24]。此方法的主要思路是先通过混合物的校正矩阵借最小二乘法求得各组分的纯物种谱，然后再利用如此求得的纯物种谱去求得未知待测混合物的各组分的浓度，这样做有以下两个好处：① 因各组分的纯物种谱是由混合物的校正矩阵求得，故有一定的抗非线性功效；② 因校正矩阵可由自己确定，故可很方便进行实验设计，这样由实验设计所估计的未知样本的各组分浓度可望更准确，可排除一些意外样本（奇异样本）的影响。K-矩阵法的基本数学模型由 $\boldsymbol{Y}=\boldsymbol{KC}$ 表示，为一线性模型，如在光谱定量分析中，就可看成是 Lambert-Beer 定律，$\boldsymbol{Y}=\boldsymbol{KC}$ 中的 $\boldsymbol{K}$ 实际上就是纯物质的标准光谱量测矩阵，它的每一列就是一个物种的标准谱。

【基本算法】

K-矩阵法的计算步骤如下：

（1）用最小二乘法求出 $\boldsymbol{K}$，即

$$K=\boldsymbol{YC}^{t}(\boldsymbol{CC}^{t})^{-1} \tag{6-33}$$

或者

$$\boldsymbol{K}^{t}=(\boldsymbol{CC}^{t})^{-1}\boldsymbol{CY}^{t} \tag{6-34}$$

（2）用求得的 $\boldsymbol{K}$ 借最小二乘法求出未知混合体系的浓度矢量 $\boldsymbol{c}_{未知}$ 或浓度矩阵 $\boldsymbol{C}_{未知}$：

$$\boldsymbol{c}_{未知}=(\boldsymbol{K}^{t}\boldsymbol{K})^{-1}\boldsymbol{K}^{t}\boldsymbol{y}_{未知} \tag{6-35a}$$

或

$$\boldsymbol{C}_{未知}=(\boldsymbol{K}^{t}\boldsymbol{K})^{-1}\boldsymbol{K}^{t}\boldsymbol{Y}_{未知} \tag{6-35b}$$

这第二步实际就是在第五章讲的直接校正的多元回归方法。从上述计算步骤可以看出 K-矩阵法要进行两次最小二乘，即需两次求逆，这样就难免使得计算误差变大。如果同样是采用间接校正的方法，但只求一次逆以减少误差的传递扩大，这就产生了所谓的 P-矩阵法[25]。

（二）*P*-矩阵法

【基本思路】

P-矩阵法是以 $C=PY$ 为其数学模型，直接以校正矩阵的浓度阵为预测目标，求得回归系数矩阵 P^t，直接从混合物量测矩阵就可求出混合物的组分浓度，这样就避免了两次求逆过程，简化了 *K*-矩阵法。

【基本算法】

P-矩阵法的计算过程如下：

（1）用最小二乘法求出 P^t

$$P_{(n\times m)}=C_{(n\times p)}Y^t_{(m\times p)}(Y_{(m\times p)}Y^t_{(m\times p)})^{-1} \tag{6-36}$$

或者

$$P^t_{(n\times m)}=(Y_{(m\times p)}Y^t_{(m\times p)})^{-1}Y_{(m\times p)}C^t_{(n\times p)} \tag{6-37}$$

（2）再用求得的 P 直接就可计算未知混合体系的浓度矢量 $c_{未知}$ 或浓度矩阵 $C_{未知}$：

$$c_{未知}=P_{(n\times m)}y_{未知} \tag{6-38}$$

或

$$C_{未知}=P_{(n\times m)}Y_{未知} \tag{6-39}$$

这样类似于 *K*-矩阵法的计算过程似乎不存在问题。可是，*P*-矩阵法却不同于 *K*-矩阵法，上述过程对 *K*-矩阵法行得通，而对于 *P*-矩阵法却的确存在问题。*K*-矩阵法是对浓度矩阵的协方差阵（CC^t）求逆，这个协方差阵是一个（$n\times n$）阶的矩阵，是满秩矩阵。然而，*P*-矩阵法是对混合物量测矩阵的协方差阵（YY^t）求逆，而它是一个（$m\times m$）阶的矩阵，为一奇异矩阵。解决的办法是只从 m 个分析通道（或为波长点）选出 n 个来组成新的混合物量测矩阵 Y，以保证其协方差阵满秩。这样做不仅有可能丢失很多有用信息，而且，这 n 个分析通道亦很难选出（有关这样的最佳波长选取方法可参看文献[12]）。正如在对多元线性回归进行讨论时指出的那样，只要有关量测误差确为高斯正态分布且具有零均等方差，则量测点越多，估计浓度的方差越小，也就是说，在一般情况下，只要不引进非线性量测点，应该是量测点越多越好。所以，以上讨论的有关 *P*-矩阵法的这一局限是一个很值得研究的弱点，就是对这一弱点的克服而产生著名的主成分回归法和偏最小二乘法。

（三）**主成分回归法**

【基本思路】

主成分回归法实际上可以看成是对 *P*-矩阵法的一种改进。它不但可尽可能多地保持有用信息，即保留所有的线性量测点，而且还保持了 *P*-矩阵法的既可进行实验设计又可一步求解的优点。它采用多元统计中的主成分分析方法，先对混合物量测矩阵 Y 矩阵直接进行分解，然后只取其中的主成分来进行回归分析，故有主成分回归之称。对 Y 矩阵进行直接分解在数学上有几种方法。在化学计量学中一般采用的方法是非线性迭代偏最小二乘算法（NIPALS）[26]。这种方法实际上是沿于 Von Mises 的乘幂法[27]。另一种方法是线性代数中常用的奇异值分解法（SVD）。奇异值分解法可将任意阶实数矩阵分解成为 3 个矩阵的积，即

$$Y=USV^t \tag{6-40}$$

式中，S 为对角矩阵，它收集了 Y 矩阵的特征值，U 和 V^t 分别为标准列正交和标准行正交矩阵，收集了这些特征值所对应的列特征矢量和行特征矢量，在多元统计的主成分分析中，一般被称为得分矩阵和荷载矩阵。主成分分析的数学与几何意义示意于图 6-6。对于 *P*-矩阵的数

学模型 $C=PY$，如果能求得量测矩阵 Y 的广义逆 Y^{-}，那么，P 矩阵则可以直接求出，即 $P=CY^{+}$。

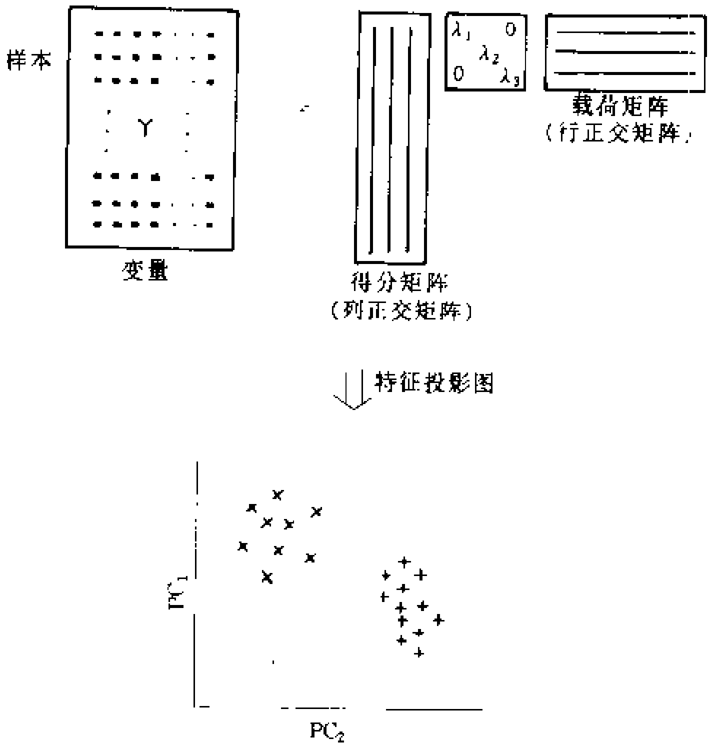


图 6-6　主成分分析数学与几何意义示意图

事实上，利用奇异值分解方法，是很容易得到任意实数矩阵的广义逆的。任何量测过程都存在着量测误差，所以混合物量测矩阵可以表示为

$$Y=Y^{0}+E \tag{6-41}$$

式中，Y^{0} 表示只含混合物的量测值和一部分植入误差的矩阵（参见本章第四节“一”），而 E 则表示误差矩阵，它由被舍弃的即很小的特征值所代表的那部分特征列矢量和特征行矢量的乘积而得，有时也称为主成分模型误差。这部分量测误差是与 Y^{0} 正交的，它与实际量测误差不完全同一。

由于已设 n 是体系的组分数，如果体系确为线性体系，则这种情况下的体系主成分数应为 n，即此时由奇异值分解算法所得的收集了 Y 矩阵的特征值的对角矩阵，应是前 n 个特征值显著大于其余的特征值，它们已包括了该矩阵的全部的化学信息，实因混合体系中化学物种数就等于该混合物量测矩阵的秩。一般只需取前 n 个特征值和特征矢量作为主成分（有非线性因素存在时，主成分数可大于 n，在化学计量学中一般用交叉校验方法来确定所需主成分数），其余部分可看成误差被丢弃。如记只取前 n 个特征值的 S 矩阵为 S^{*}，记只取前 n 个相对应的特征矢量的得分矩阵 U 为 U^{*}，荷载矩阵 V 为 V^{*}，则有

$$Y^{0}=U^{*}S^{*}V^{t*} \tag{6-42a}$$

对式（6-42）可以直接求出剔除了主成分模型误差的重构混合物量测矩阵 Y^{0} 的广义逆

$$Y^{0+}=V^{*}(S^{*})^{-1}U^{t*} \tag{6-42b}$$

容易证明，Y^{0-} 的确满足广义逆定义的 4 个条件（参见第十一章）。

求得剔除了误差的量测矩阵 Y^{0} 的广义逆 Y^{0-}，主成分回归法的计算过程就不难说明了。

【基本算法】

在此将介绍两种算法。

（1）算法Ⅰ

1）用奇异值分解法由式(6-42b)求出量测矩阵的广义逆 Y^{0-}，然后，用它求出回归系数矩阵 P

$$P = CY^{0+} = CV^{*}(S^{*})^{-1}U^{t^{*}} \tag{6-43}$$

2）在用求得的 P 直接就可计算未知混合体系的浓度矢量 $c_{未知}$ 或浓度矩阵 $C_{未知}$：

$$c_{未知} = Py_{未知}$$

或

$$C_{未知} = PY_{未知}$$

从上述讨论中，我们可以看到，主成分回归法既保持了 P-矩阵法的可进行实验设计和一步计算的优点，又克服了 P-矩阵法要丢失大部分量测信息及损失估计准确性的弱点，确为一很好的间接校正方法，近年来在化学计量学中得到了广泛的应用[28]。

（2）算法Ⅱ

1）对混合物量测矩阵 Y 进行正交分解，即

$$Y = USV^{t} = U^{*}S^{*}V^{t^{*}} = U^{*}G^{t^{*}} \tag{6-44}$$

在这里 U^{*}，S^{*}，$V^{t^{*}}$ 的意义同前，实际上，这一正交分解就相当于主成分重构，并且有 $S^{*}V^{t^{*}} = G^{t^{*}}$，而且 $G^{t^{*}}$ 仍为一行正交矩阵，只是不是标准行正交矩阵而已。

2）用 $G^{t^{*}}$ 对浓度阵 C 进行回归可得回归系数阵 R，即对 $C = RG^{t^{*}}$ 式进行最小二乘运算：

$$\begin{aligned} R &= CG^{*}(G^{t^{*}}G^{*})^{-1} = C(V^{*}S^{*})(S^{*}V^{t^{*}}V^{*}S^{*})^{-1} = C(V^{*}S^{*})(S^{*}S^{*})^{-1} \\ &= C(V^{*}S^{*})(S^{*2})^{-1} = CV^{*}(S^{*})^{-1} \end{aligned}$$

3）预测未知样本量测矩阵 $Y_{未知}$，因为

$$Y_{未知} = U^{*}G^{t^{*}}_{未知}$$

所以

$$G^{t^{*}}_{未知} = U^{*t}Y_{未知}$$

再利用 $C_{未知} = RG^{t^{*}}_{未知}$ 式，可得

$$C_{未知} = RG^{t^{*}}_{未知} = RU^{*t}Y_{未知}$$

求得 $C_{未知}$。事实上

$$\begin{aligned} C_{未知} &= RU^{*t}Y_{未知} = CV^{*}(S^{*})^{-1}U^{*t}Y_{未知} = CV^{*}(S^{*})^{-1}U^{*t}Y_{未知} \\ &= CY^{0-}Y_{未知} = CY^{0+}Y_{未知} \\ &= PY_{未知} \end{aligned}$$

可见，这两种计算过程殊途同归。

（四）偏最小二乘法

【基本思路】

主成分回归方法的主要思想是直接对混合物量测矩阵 Y 进行正交分解，继而用浓度矩阵 C 对被分解的正交主成分进行回归。偏最小二乘法则更进一步，它不但对混合物量测矩阵 Y 进行正交分解，而且在分解 Y 的同时将浓度矩阵 C 也进行正交分解，这样浓度矩阵的主成分也被分离出来：

$$Y = USV^t = U^* S^* V^{t*} + E_Y = T^* V^{t*} + E_Y \quad (6\text{-}45a)$$

$$C = PGQ^t = P^* G^* Q^{t*} + E_c = R^* Q^{t*} + E_c \quad (6\text{-}45b)$$

式中，由混合物量测矩阵 Y 分解得到的矩阵 T^* 以及由浓度矩阵 C 分解得到的矩阵 R^*，代表了除去大部分噪声后的响应和浓度的信息。而且，在进行同时分解时，考虑了矩阵 T^* 和矩阵 R^* 之间应有的线性关系，这就构成了偏最小二乘法不同于主成分回归的主要之点，即在分解矩阵 Y 时考虑矩阵 C 的因素，而在分解矩阵 C 时考虑矩阵 Y 的因素，交互校验相互影响，通过迭代时交换迭代矢量而使两个分解过程合二为一。

【基本算法】

偏最小二乘法的迭代过程是通过由 H. Wold 提出的非线性迭代偏最小二乘算法（NIPALS）而完成的。具体计算步骤如下：

1）取浓度阵 C 的某一列，记为 r，作分解的初始迭代矢量。

2）以 r 代替 T^* 矩阵的分解矢量 t 来计算 v^t（V^{t*} 矩阵的一行）：

$$v^t = r^t Y / (r^t r)$$

（注意：以上两步等于是用浓度阵 C 的某一列作为初始迭代矢量来进行 Y 矩阵的正交分解，即分解 Y 矩阵时利用浓度阵 C 的信息。）

3）对所求矢量 v^t 进行标准化（归一化）处理：$v^t_{新} = v^t_{旧} / \| v^t_{旧} \|$

4）由 v^t 来计算矢量 t（T^* 矩阵的一列）：$t = Yv/(v^t v)$

5）以 t 代替 R^* 矩阵的分解矢量 r 来计算 q^t（Q^{t*} 矩阵的一行）：

$$q^t = t^t C / (t^t t)$$

（注意：这一步等于是用 Y 矩阵的分解矢量 t 来作为 C 矩阵的正交分解矢量 r 以得到其另一正交分解矩阵（Q^{t*} 矩阵）的一行。这说明分解浓度阵 C 时也利用了矩阵 Y 的信息。）

6）对所求矢量 q^t 进行标准化（归一化）处理：$q^t_{新} = q^t_{旧} / \| q^t_{旧} \|$

7）由 q^t 来计算矢量 r（R^* 矩阵的一列）：$r = Xq/(q^t q)$

8）判断 t 是否收敛，即 $\| t_{前一轮} - t_{后一轮} \|$ 是否小于某一给定常数 θ。如收敛则开始下一步主因子分解，否则，返回到第二步，继续进行主因子分解迭代。此时返回第二步应是利用第七步所求的矢量 r 来计算了。

这样计算出来的 t 值已失去正交性质，可以引入补充的计算环节使之保持正交，但这对本书所讨论的多元校正的问题无关紧要。上述计算得到的 q，v^t 应予保留，供下一步测试时用。如要求在测试时检验校正模型是否正确，则 r，t 也需保留。

下一步计算 r 与 t 之间的内部关系，进行回归运算：

$$b = r^t t / (t^t t)$$

以上计算过程可示意地用图 6-7 表示。值得指出的是，以上计算的各分解矢量均对应第一个主因子，即 r_1，q^t_1，t_1，v^t_1，为简洁起见，在上述过程中下标均略去。下一步计算残差阵：

$$E_{Y,1} = Y - t_1 v^t_1$$

$$E_{C,1} = C - r_1 q^t_1 = C - b_1 t_1 q^t_1$$

以 $E_{Y,1}$ 代替 Y，以 $E_{C,1}$ 代替 C。返回上述的主因子迭代过程，计算下一个主因子，得 r_2，q^t_2，t_2，v^t_2 又可计算下一步残差阵 $E_{Y,2}$ 和 $E_{C,2}$，依次类推，直至找出 Y 和 C 的全部主因子为止。主因子的数目可用交互校验法或方差比较来决定。

对于未知试样，由其量测的矢量 $y_{未知}$ 或其量测的矩阵 $Y_{未知}$，利用 $Y=T^{*}V^{t^{*}}$ 的关系及在校正步骤中存储的 $V^{t^{*}}$，可算出 $t_{未知}$，继由 t 与 r 的关系可求出 $r_{未知}$，由存储的 $Q^{t^{*}}$，即可求出 $c_{未知}$。

从上述讨论可知，间接校正的 K-矩阵、P-矩阵、主成分回归以及偏最小二乘法是一脉相通、相互连贯的，从中可以明显地看出一个逐步发展的过程。P-矩阵方法克服了 K-矩阵方法要求两次逆而引起误差扩大的缺点；而主成分回归则克服了 P-矩阵不满秩求逆或需丢失光谱信息的弱点，采用主成分分解 Y 矩阵（量测矩阵）以得其广义逆而显著地改善了 P-矩阵方法；偏最小二乘法则不但立足于分解 Y 矩阵，而且还同时分解 C 矩阵（浓度校正矩阵），而且在分解 Y 矩阵时考虑 C 矩阵的因素，在分解 C 矩阵时考虑 Y 矩阵的因素，加强双方回归对应计算的关系，以求获得最佳回归预测的效果。

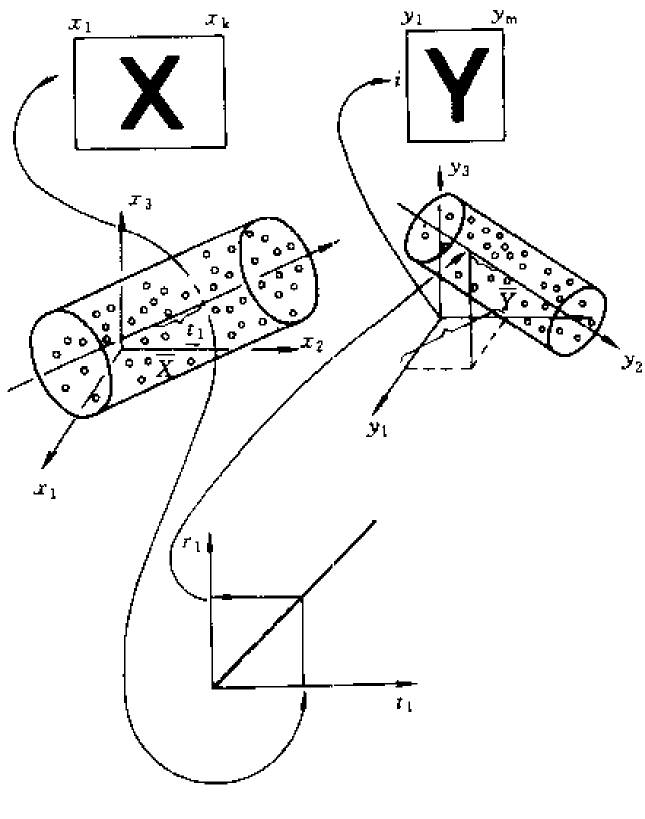


图 6-7　PLS 投影及内相关示意图

由于主成分回归与偏最小二乘回归对 K-矩阵和 P-矩阵方法给出了显著性的改进，故在近年来得到了相当广泛的应用，而且，在方法本身的理解和改进上，广大化学计量学家也进行了大量的工作，其中较著名的有 Hoskuldsson 对偏最小二乘法的解释和说明，他证明了偏最小二乘法本质上是对 $Y^{t}CC^{t}Y$ 或 $C^{t}YY^{t}C$ 的分解[29]，Manne[30]从算法的角度说明了，相对于主成分回归的是对 Y 矩阵进行对角分解，则 PLS1 就相当于对 Y 矩阵进行双对角分解，而 PLS2 就相当于对 Y 矩阵进行三对角分解。

现将多元间接校正方法的计算公式分列于表 6-3，以更简洁的形式说明它们之间的关系。

表 6-3　间接多元校正方法的矩阵计算比较

方　法	K-矩阵法		P-矩阵法
数学模型	$Y=KC$		$C=PY$
校正	$K=YC^{t}(CC^{t})^{-1}$		$P=CY^{t}(YY^{t})^{-1}$
	$K^{t}=(CC^{t})^{-1}CY^{t}$		$P^{t}=(YY^{t})^{-1}YC$
预测	$c_{未知}=(K^{t}K)^{-1}K^{t}y_{未知}$		$c_{未知}=P_{(n\times m)}y_{未知}$
	$C_{未知}=(K^{t}K)^{-1}K^{t}Y_{未知}$		$C_{未知}=P_{(n\times m)}Y_{未知}$
	PCR	**PLS2**	**PLS1**
	$Y=U^{*}S^{*}V^{*t}$	$Y=URW^{t}$	$Y=URW^{t}$
	$Y^{+}=V^{*}(S^{*})^{-1}U^{*t}$	$Y^{+}=WR^{-1}U^{t}$	$Y^{+}=WR^{-1}U^{t}$
	$P=CY^{+}$	$P=CY^{-}$	$P=cY^{+}$
	S^{*}：对角矩阵	R：三对角矩阵	R：二对角矩阵

三、通用标准加入法

经典分析化学中经常采用标准加入法，以克服基体或底液对分析校正的影响。事实上，标准加入的方法同样可推广到多元校正的情况。前两节讨论的多元线性回归和 Kalman 滤波算法都是首先要获得校正所需的敏感度矩阵或是各物种的标准谱，如果事先无法获得标准谱，即由于待测样本的基体条件不清楚时，无法找到标准谱的量测条件时，即可采用通用标准加入法[31,32]来进行多元校正。

【基本思路】

通用标准加入法的实质是在待测多组分体系中直接加入多种待测标准物，以构成一个新的量测矩阵，再从这新的量测矩阵求得在此基体溶液中的各标准物的标准量测谱，继而用求得的标准谱对待测多组分体系进行直接校正，即可求得此多组分体系的各组分的浓度来。

【基本过程和算法】

设有一待测试样 $\boldsymbol{y}_0$，其 n 个组分的浓度矢量为 $\boldsymbol{c}_0$，向此试样中进行 m 次标准加入（未作加入的原始试样认作加入量为零的试样），这样可得一新的量测矩阵 $\boldsymbol{Y}$，有

$$\boldsymbol{Y}_{p\times m}=\boldsymbol{X}_{p\times n}\boldsymbol{C}_{n\times m}+\boldsymbol{E}_{p\times m} \tag{6-46}$$

式中

$$\boldsymbol{Y}=[\boldsymbol{y}_1,\ \boldsymbol{y}_2,\ \cdots,\ \boldsymbol{y}_m]=\begin{bmatrix} y_{11} & y_{21} & y_{31} & \cdots & y_{m1} \\ y_{12} & y_{22} & y_{32} & \cdots & y_{m2} \\ y_{13} & y_{23} & y_{33} & \cdots & y_{m3} \\ \vdots & \vdots & \vdots & & \vdots \\ y_{1p} & y_{2p} & y_{3p} & \cdots & y_{mp} \end{bmatrix} \tag{6-47}$$

式中，$\boldsymbol{y}_i(i=1,2,\cdots,m)$表示第 i 次加入时对混合物进行量测时所得的量测矢量，每次量测都是在 p 个分析通道或波长上进行的。一般要求 $m>n$。在此，浓度阵 $\boldsymbol{C}$

$$\boldsymbol{C}=\boldsymbol{C}_0+\Delta\boldsymbol{C}=\begin{bmatrix} c_1 & c_1 & c_1 & \cdots & c_1 \\ c_2 & c_2 & c_2 & \cdots & c_2 \\ c_3 & c_3 & c_3 & \cdots & c_3 \\ \vdots & \vdots & \vdots & & \vdots \\ c_n & c_n & c_n & \cdots & c_n \end{bmatrix}+\begin{bmatrix} \Delta c_{11} & \Delta c_{21} & \Delta c_{31} & \cdots & \Delta c_{m1} \\ \Delta c_{12} & \Delta c_{22} & \Delta c_{32} & \cdots & \Delta c_{m2} \\ \Delta c_{13} & \Delta c_{23} & \Delta c_{33} & \cdots & \Delta c_{m3} \\ \vdots & \vdots & \vdots & & \vdots \\ \Delta c_{1n} & \Delta c_{2n} & \Delta c_{3n} & \cdots & \Delta c_{mn} \end{bmatrix} \tag{6-48}$$

式中，Δc_{ij}表示第 i 次对第 j 个待测组分加入的浓度。注意在加入时，必须使原待测样本的溶液的体积保持不变（即加入小体积的较浓标准溶液）或采用稀释至相同体积的方法，这样，$\boldsymbol{C}_0$ 可看成是一个每列相同的矩阵，即每列都由 $\boldsymbol{c}_0$ 构成。式（6-46）中的 $\boldsymbol{X}$ 矩阵仍为标准谱矩阵，它的每一列皆由体系中各组分的标准量测谱构成，即

$$\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]=\begin{bmatrix} x_{11} & x_{21} & & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ x_{13} & x_{23} & \cdots & x_{n3} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1p} & x_{2p} & \cdots & x_{np} \end{bmatrix} \tag{6-49}$$

如果用 y_0 来组成一个与 $\boldsymbol{Y}$ 相同大小的矩阵 $\boldsymbol{Y}_0$，即

$$Y_0=[y_0,y_0,\cdots,y_0]=\begin{bmatrix} y_{01} & y_{01} & y_{01} & \cdots & y_{01} \\ y_{02} & y_{02} & y_{02} & \cdots & y_{02} \\ y_{03} & y_{03} & y_{03} & \cdots & y_{03} \\ \vdots & \vdots & \vdots & & \vdots \\ y_{0p} & y_{0p} & y_{0p} & \cdots & y_{0p} \end{bmatrix}$$

矩阵 Y_0 中包括 m 列相同的矢量 y_0，则有，

$$\Delta Y_{p\times m}=Y-Y_0=X(C_0+\Delta C)-X(C_0)=X_{p\times n}\Delta C_{n\times m} \tag{6-50}$$

式(6-50)中矩阵 $\Delta Y_{p\times m}$和 $\Delta C_{n\times m}$皆为已知，采用最小二乘法即可容易求出 $X_{p\times n}$来，

$$X=\Delta Y\Delta C^{t}(\Delta C\Delta C^{t})^{-1}$$

求得标准谱矩阵 X，该体系的未知浓度矢量 c_0 就可用下式立即求得

$$c_0=(X^{t}X)^{-1}X^{t}y_0 \tag{6-51}$$

在此值得提出的是，ΔC 是分析过程中人为作 m 次加入的标准物浓度构成的矩阵，这个矩阵可以由分析者进行设计。合理的设计应当使各次加入的标准浓度线性无关，为此可采用因子或正交设计来实现。

四、稳健多元校正方法

在前几节的讨论中，我们知道，在以上的几种多元校正的模型中，一般都假设化学的量测误差都是服从正态分布的等方差白噪声，但是在实际分析工作中这一假设不一定满足。Clancey 对 250 次量测包括 50000 金属化学分析的误差分布进行调查的结果表明，其中只有 10%～15%可以看成是服从正态分布[33]，在对血样的调查也给出了相类似的结果[34]。出现这样的调查结果的原因可归结为两个原因：① 化学量测的误差本来就不一定服从正态分布；② 在量测过程中存在有奇异点。所谓奇异点是指远离分布整体的量测值。产生奇异点的原因很多，它可以是过失差错，也可能是样本点没有落在实验设计的范围之内，也可能就是极少数就来自此分布的奇异点。Barnett 和 Lewis 对此作出了较详细的讨论[35]。稳健多元分析的方法正是为消除或减轻奇异点影响和直接对非正态分布数据进行解析而设计的。

很多用于化学数据解析的方法都是基于最小二乘估计（LS），如主成分分析（PCA），多元线性回归（MLR）主成分回归（PCR）和偏最小二乘（PLS）方法都属于此类。然而，最小二乘方法是不稳健的，这可能是由其目标函数是用残差的平方和所决定的：

$$\mathrm{Min}\sum r_i^2=\mathrm{Min}\sum(y_i-a_1x_{i1}-a_2x_{i2},\cdots,-a_mx_{im})^2 \tag{6-52}$$

式中，r_i 是第 i 点的残差；$y_i(i=1,2,\cdots,n)$为第 i 点的量测值(因变量)；$x_{ij}(i=1,2,\cdots,n;j=1,2,\cdots,m)$为第 i 点量测点上的第 j 个自变量；$a_j(j=1,2,\cdots,m)$为第 j 个最小二乘估计值。从式（6-52）可以看出，如某一个量测值 y_i 为一奇异点，它的残差很大，再经平方运算，则对最小二乘估计将产生很大影响。故稳健估计的第一步首先是由 Edgeworth[36]在 1887 年提出的最小一乘估计，即

$$\mathrm{Min}\sum|r_i| \tag{6-53}$$

然而，最小一乘估计只对 y_i 的奇异点稳健，而对另一类奇异点，亦称为坏杠杆点的值非常敏感。所谓杠杆点在回归分析中称为强影响点[37]，有关这类奇异点的例子可参阅文献[38]。为评价方法的稳健性，Hodges[39]引入了崩溃点的概念，这一概念的更一般的计算公式则由 Hampel[40]给出。一般说来，崩溃点的概念可解释为强烈影响估计偏离其“实际”情况的最小

的奇异点数与估计时所包含点数的比值。最小二乘估计的崩溃点是$1/n$。这就是说，只要在n个量测点中有一个是奇异点，它就可以完全破坏最小二乘估计。值得提出的是，如果存在坏杠杆点，最小一乘的崩溃点也是$1/n$！

在统计学中，一般用两种方法来处理含奇异点和非正态分布的数据，即诊断方法和稳健估计方法。如 Rousseeuw 和 leory 指出的那样[41]，这两种方法虽处理顺序完全相反，但却殊途同归，具有相同目的。诊断方法是着重于首先找出奇异点，在剔除这些奇异点后继用经典估计方法处理；而稳健估计方法则是在解析过程中找到数据分布的主体，构造稳健估计模型，同时使数据中的奇异点不对估计产生影响（尽管这些奇异点有很大的残差）。在实际的应用中，两种方法几乎产生同样的结果，选择哪种方法来进行稳健估计无关紧要。在这里我们对近年来在统计学和化学计量学中出现的稳健方法进行讨论，使读者对多元分析中的稳健方法有一全面性了解。

（一）稳健回归方法

稳健回归方法是稳健估计方法中一种最常用的方法，其主要思路是在回归过程中找到数据分布的主体，构造稳健回归估计模型，在进行估计同时使数据中的奇异点不对估计产生影响（尽管这些奇异点有很大的残差），则由此而得的估计是稳健的。

1. 最小一乘法

【基本思路】

最小一乘法首先是由 Edgeworth[36]在 1887 年提出，其目标函数为 $\mathrm{Min}\sum|r_i|$，以减小最小二乘由于对残差进行平方而增大异常点对拟合的影响，从而达到稳健的效果。由于绝对值的计算不便于微分，最小一乘很难得到像最小二乘那样的简洁解析解，一般需采用优化方法求解，本文介绍的是一种基于数论的序贯全局优化方法（参见第四章）[42]。该优化方法采用数论方法在搜索区间均匀布点，继序贯收缩搜索空间，故有序贯数论优化算法（SNTO）之称。与其他全局优化算法比，SNTO 具有原理清晰明了，算法易于实现等特点。当然，最小一乘法也可用线性规划的方法来求解[43,44]，读者可参阅文献[45]。

【基本算法】

SNTO 算法的第一步就是确定搜索区间，继在搜索区间均匀布点，序贯收缩以达到最优点的估计。

(1) 多元校正浓度估计的约束区间

从化学知识可知，浓度估计值 $c_i(i=1,\cdots,n)$都大于零，即 SNTO 算法的搜索下界为 $\boldsymbol{a}=(0,\cdots,0)$，另从式(6-1)和式(6-2)可知

$$y_i=c_1x_{i1}+\cdots+c_nx_{in}+e_i \tag{6-54}$$

对于任一组分 k，有

$$y_i>c_kx_{ik}+e_i$$

亦即

$$c_k\leqslant(y_i+\alpha)/x_{ik}\leqslant\underset{i\in\{1,\cdots,m\}}{\mathrm{Max}}[(y_i+\alpha)/x_{ik}]\quad(k=1,\cdots,n) \tag{6-55}$$

式中，m 为量测点数（波长数），α 为一正数，以替代误差的作用，其余符号意义同前。所以，多元校正的搜索区间的上界为

$$b=(\mathrm{Max}[(y_i+\alpha)/x_{i1}],\cdots,\mathrm{Max}[(y_i+\alpha)/x_{in}])$$

(2) n 维实数空间中的长方体域 $[\boldsymbol{a},\boldsymbol{b}]$ 时的 SNTO 算法

令 $\boldsymbol{a}=(0,\cdots,0)$，$\boldsymbol{b}=(\mathrm{Max}[(y_i+\alpha)/x_{i1}],\cdots,\mathrm{Max}[(y_i+\alpha)/x_{in}])$，以及 $\boldsymbol{x}=(x_1,\cdots,x_s)$，SNTO 的过程如下：

① 初始化。令 $t=0$，$D^{(0)}=D$，$\boldsymbol{a}^{(0)}=\boldsymbol{a}$ 以及 $\boldsymbol{b}^{(0)}=\boldsymbol{b}$。

② 产生 *NT-net*。用 glp 集产生在 $D^{(t)}=[\boldsymbol{a}^{(t)},\boldsymbol{b}^{(t)}]$ 上均匀分布含 n_t 点的点集 $\rho^{(t)}$。

③ 计算近似最优点。确定满足下式的 $\boldsymbol{x}^{(t)}\in\rho^{(t)}\cup\{\boldsymbol{x}^{(t-1)}\}$ 和 $M^{(t)}$：

$$M^{(t)}=\mathrm{obj}(\boldsymbol{x}^{(t)})=\sum|r_i^{(t)}|\leqslant\mathrm{obj}(\boldsymbol{x}),\forall\boldsymbol{x}\in\rho^{(t)}\cup\{\boldsymbol{x}^{(t-1)}\},$$

式中，$\boldsymbol{x}^{(-1)}$ 是空集，$\boldsymbol{x}^{(t)}$ 和 $M^{(t)}$ 是到目前为止 $\boldsymbol{x}^*$ 和 M 的最佳近似。

④ 终止准则。令 $\boldsymbol{h}^{(t)}=(\boldsymbol{b}^{(t)}-\boldsymbol{a}^{(t)})/2$。如果 $\max(\boldsymbol{h}^{(t)})<\delta$，$\delta$ 为一预置的较小数，则可认为 $D^{(t)}$ 收缩到足够小，终止算法并接受 $\boldsymbol{x}^{(t)}$ 和 $M^{(t)}$。否则，执行下一步。

⑤ 收缩搜索域。构造新的搜索域 $D^{(t+1)}=[\boldsymbol{a}^{(t+1)},\boldsymbol{b}^{(t+1)}]$：

$$\boldsymbol{a}_i^{(t+1)}=\max(\boldsymbol{x}_i^{(t)}-\gamma\boldsymbol{c}_i^{(t)},\boldsymbol{a}_i)$$

且

$$\boldsymbol{b}_i^{(t+1)}=\min(\boldsymbol{x}_i^{(t)}+\gamma\boldsymbol{c}_i^{(t)},\boldsymbol{b}_i)$$

式中，γ 是一预置的收缩因子。令 $t=t+1$，转到②。

方开泰等建议选取 $n_1>n_2=n_3=\cdots$ 并使用 $\gamma=0.5$。此外，虽然以上算法是为长方体搜索域中的优化问题而设计的，但可很容易地将它推广至一般的搜索域中。

2. *M* 估计法

【基本思路】

M 估计是极大似然估计的一种简称[46]。其主要思路也是改变估计的目标函数。反最小二乘估计用残差平方和为目标函数的做法，它重新定义了一个目标函数

$$\mathrm{Min}\sum\rho(r_i)$$

式中，$\rho(\cdot)$ 为一偶函数，即 $\rho(-t)=\rho(t)$，且在 t 等于零时具有唯一的最小。对式(6-56)的回归参数求导可得

$$\sum\Psi(r_i/s)\boldsymbol{x}_i=\boldsymbol{0} \tag{6-56}$$

式中，$\Psi(\cdot)$ 为 $\rho(\cdot)$ 的导函数；s 是残差 r_i 的分散度的估计值。式(6-57)中的两个矢量的定义分别为：$\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{im})^{\mathrm{t}}$ 和 $\boldsymbol{0}=(0,0,\cdots,0)^{\mathrm{t}}$。如果选 $\rho(r_i)=1/2r_i^2$，则 *M* 估计就是经典的最小二乘估计。

实际上，*M* 估计是直接从构造 Ψ 函数而不是构造 ρ 函数出发，不同的 *M* 估计有着不同的 Ψ 函数。在这里给出一个例子来加以说明。Andrews[47]定义了如下的 Ψ 函数：

$$\Psi(z)=\begin{cases}\sin(z/c) & |z|<c \quad (6\text{-}57\mathrm{a})\\ 0 & |z|>c \quad (6\text{-}57\mathrm{b})\end{cases}$$

式中，c 是一个截断参数。z 可以是残差也可以是残差的函数，所以 Ψ 可以看成是回归参数 $\alpha_j(j=1,2,\cdots,m)$ 的函数，这样，只要得到了 α_j，就可以得到 Ψ。因此，*M* 估计在实际的计算中是一个迭代过程[48]：

【基本算法】

(1) 最小二乘估计得到回归参数的初始估计 $\boldsymbol{\alpha}=(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$；然后，用下式对所得的初始残差进行处理，目的在于降低大残差对下一步对回归参数 $\alpha_j(j=1,2,\cdots,m)$ 估计的影响：

$$r_{i新}=\begin{cases}1.5\times\mathrm{median}(r_i)\times\mathrm{sign}(r_i) & |r_i|>1.5\times\mathrm{median}(r_i) \quad (6\text{-}58\mathrm{a})\\ r_i & |r_i|<1.5\times\mathrm{median}(r_i) \quad (6\text{-}58\mathrm{b})\end{cases}$$

式中，median(r_i)表示取残差的中位数，它是一个替代均值估计的稳健指标。回归参数的初始估计即可由式(6-59)得到。

$$\alpha^{\text{ini}}=\alpha+\Delta\alpha \tag{6-59}$$

式中

$$\Delta\alpha=(\boldsymbol{X}^{\text{t}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\text{t}}\boldsymbol{r}_{\text{新}}$$

(2) 迭代过程

令

$$(w_i^{(k+1)})^2=\Psi(r_i^{(k)}/s^{(k)})/(r_i^{(k)}/s^{(k)})$$

式中

$$s^{(k)}=\text{median}(|r_i^{(k)}|)$$

这样，方程(6-56)可变为

$$\sum[r_i^{(k)}\Psi(r_i^{(k)}/s^{(k)})\boldsymbol{x}_i/r_i^{(k)}]s^{(k)}=\boldsymbol{0}s^{(k)} \tag{6-60}$$

将式(6-60)换成用 $w_i^{(k+1)}$ 表示，即得

$$\begin{aligned}\sum[r_i^{(k)}\Psi(r_i^{(k)}/s^{(k)})\boldsymbol{x}_i/r_i^{(k)}]s^{(k)} &= \sum r_i^{(k)}[\Psi(r_i^{(k)}/s^{(k)})\boldsymbol{x}_i/(r_i^{(k)}/s^{(k)})]\\ &= \sum r_i^{(k)}[\Psi(r_i^{(k)}/s^{(k)})/(r_i^{(k)}/s^{(k)})]\boldsymbol{x}_i = \sum r_i^{(k)}[(w_i^{(k+1)})^2]\boldsymbol{x}_i\\ &= \sum\{(w_i^{(k+1)})[y_i-(\boldsymbol{\alpha}^{(k+1)})^{\text{t}}\boldsymbol{x}_i]\}(w_i^{(k+1)})\boldsymbol{x}_i\\ &= \sum(w_i^{(k+1)})[y_i-(\boldsymbol{\alpha}^{(k+1)})^{\text{t}}\boldsymbol{x}_i]=\boldsymbol{0}\end{aligned} \tag{6-61}$$

解式(6-61)可得

$$\boldsymbol{\alpha}^{(k+1)}=[\boldsymbol{X}^{\text{t}}(\boldsymbol{W}^{(k+1)})^2\boldsymbol{X}]^{-1}\boldsymbol{X}^{\text{t}}(\boldsymbol{W}^{(k+1)})^2\boldsymbol{y} \tag{6-62}$$

式(6-62)实际就是加权最小二乘估计的解析解，在这里$(\boldsymbol{W}^{(k+1)})^2$为一对角矩阵，其对角元素就是 $\Psi(r_i^{(k)}/s^{(k)})/(r_i^{(k)}/s^{(k)})$。所以，在实际的计算过程中，就是通过式(6-57)得到 $w_i^{(k+1)}$，然后用式 (6-62) 得到 $\boldsymbol{\alpha}^{(k+1)}$，再求得新的残差以开始下一轮迭代。重复此迭代过程，直至 $\boldsymbol{\alpha}^{(k+1)}$与 $\boldsymbol{\alpha}^{(k)}$的差值的模小于某一给定值。表 6-4 给出了一个这样的计算例子。从这一例子可以看到，*M* 估计的收敛速度很快。由式(6-58a)～(6-59)定义的初始残差和初始回归参数估计就排除了三个奇异点的影响（即表 6-4 中的三个用下划线标明的数值：1.704，1.792 和 2.372）。在迭代过程中，第二次迭代就找到了还余下的那个奇异点（y_i=1.006，其残差在第二次迭代中为 0.2778）。在第三次的迭代过程中，它的影响亦被消除。此时除 4 个奇异点外，所有的权因子 $w_i^{(3)}$ 都变成了 1。

表 6-4　*M* 估计迭代过程

迭代次数			第一次		第二次		第三次	
原始数据			α_1	α_2	α_1	α_2	α_1	α_2
			1.336	0.3981	0.7033	0.2086	0.5005	0.4000
y	x_1	x_2	残差	权重	残差	权重	残差	权重
0.5266	0.5207	0.6640	0.0933	0.9986	0.0218	0.9999	0.0004	1.000
0.6596	0.6640	0.8185	0.0985	0.9984	0.0219	0.9999	−0.0001	1.000
0.8001	0.8185	0.9766	0.0955	0.9984	0.0207	0.9999	−0.0001	1.000
0.9399	0.9766	1.129	0.0842	0.9988	0.0175	0.9999	−0.0003	1.000
1.107	1.129	1.265	0.0675	0.9992	0.0129	1.000	−0.0001	1.000
1.183	1.265	1.375	0.0392	0.9997	−0.0012	1.000	0.0004	1.000
1.268	1.375	1.449	0.0068	1.000	−0.107	1.000	−0.0002	1.000
1.318	1.449	1.482	−0.0286	0.9999	−0.0004	1.000	−0.0004	1.000

续表

迭代次数			第一次		第二次		第三次	
原始数据			α_1	α_2	α_1	α_2	α_1	α_2
			1.336	−0.3981	0.7033	0.2086	0.5005	0.4000
y	x_1	x_2	残差	权重	残差	权重	残差	权重
1.329	1.482	1.469	0.0663	0.9993	−0.0283	0.9999	0.0001	1.000
1.299	1.469	1.409	−0.1037	0.9982	−0.0361	0.9998	−0.0003	1.000
1.227	1.409	1.307	−0.1351	0.9970	−0.0411	0.9997	0.0002	1.000
1.122	1.307	1.169	−0.1598	0.9958	−0.0440	0.9997	0.0003	1.000
0.9890	1.169	1.009	−0.1704	0.9952	0.0115	0.9997	0.0000	1.000
0.8396	1.009	0.8370	−0.1732	0.9950	−0.0428	0.9998	0.0004	1.000
1.006	0.8370	0.6671	0.1513	0.9939	0.2778	0.000	0.3199	0.000
0.5378	0.6671	0.5100	−0.1486	0.9963	0.0378	0.9998	−0.0001	1.000
1.704	0.5100	0.3735	1.137	0.000	1.268	0.000	1.299	0.000
1.792	0.3735	0.2618	1.369	0.000	1.474	0.000	1.500	0.000
2.372	0.2618	0.1754	2.071	0.000	2.151	0.000	2.171	0.000
0.1328	0.1754	0.1122	−0.0924	0.9991	0.0139	1.000	0.0002	1.000

① 最小二乘估计结果：2.3759，−1.4033；经修正后的初始估计：1.3358，−0.3981；实际值：0.5000，0.4000。

【讨论】

在化学计量学的研究中，Philips 和 Eyring 最早将 M 估计用于回归分析[49]。他们对 38 套至少含 10 个数据点以上的数据来估计一元回归参数，即回归直线的斜率和截距，发现 M 估计优于或至少与经典最小二乘估计相当。Wolters 和 Kateman 应用另一种 M 估计方法通过 Monte Carlo 模拟对不同误差分布对参数估计的影响作出了详细的研究[50]。他们的结论是：对于所含数据点大于 10 的数据，M 估计一般都优于最小二乘估计。魏万之等用 Andrews 所提函数的 M 估计来进行多组分体系解析[51]，他们的研究结果表明，如果误差分布不属于正态分布，则稳健估计明显优于最小二乘估计。谢玉珑等[52]用几种 M 估计方法来处理多元校正中的部分非线性问题，那些偏离线性模型的波长点就作为奇异点看待，他们的结果表明稳健方法对消除部分非线性数据有显著效果。

3. 最小中位平方法

【基本思路】

一般说来，如果存在坏杠杆点，最小一乘的崩溃点是 $1/n$，而 M 估计的崩溃点难于大于 30%。为了得到具有高崩溃点的稳健估计，Rousseeuw 提出了最小中位方差估计（LMS）[53]。它的目标函数是

$$\text{Min median}(r_i^2) \tag{6-63}$$

式(6-63)中 median(r_i^2)表示残差平方的中位数。LMS 估计的崩溃点可以达到 50%，这是稳健估计方法的可能达到的最高值。另外，LMS 估计对量测值的奇异点和高杠杆点都是稳健的，这样的例子可参见文献[37]。然而，LMS 估计的收敛速率较慢。一般说来，LMS 方法的计算时间与量测样本的点数（n）的立方成正比。为此，Steiger 和 Steel 提出了一个改进算法，其计算速度与 $(n\lg(n))^2$ 成正比[54]。Massart 等首次将 LMS 估计引入化学计量学[55]。Rutan 和Carr 将 LMS 估计与自适应 Kalman 滤波方法用于消除一维数据的奇异点的情况进行了比较[56]。Ukkelberg 和 Borgen 为检出奇异点基于 LMS 估计提出了一种稳健的交替回归方法[57]。

【基本算法】

该算法与前述的最小一乘完全一致，只需将最小一乘的目标函数 $\text{Min}\sum|r_i|$ 改为 Min median(r_i^2)即可。

4. 最小修剪平方法

为克服LMS估计的较慢收敛速率之不足，Rousseeuw[41]提出了最小方差修剪估计，其目标函数为

$$\mathrm{Min}\sum_{i=1}^{h}\sum(r_i^2)_{i:n} \tag{6-64}$$

式中，$(r_i^2)_{1:n}<(r_i^2)_{2:n}<\cdots<(r_i^2)_{n:n}$。

从式(6-64)可以看出，此法与最小二乘类似，不同的地方只是LTS估计只取了一部分残差较小的来进行回归（即从 $(r_i^2)_{1:n}$到 $(r_i^2)_{h:n}$）。当取 h 为 n 的一半时，LTS估计的崩溃点也可达到50%，但是它的收敛速度明显加快。

【基本算法】

该算法亦可使用前述的优化方法来实现，只需将目标函数改为 $\mathrm{Min}\sum_{i=1}^{h}(r_i^2)_{i:n}$ 即可，故此从略。

5. 稳健主成分回归法

最近，由Walczak和Massart提出了一个稳健主成分回归法（RPCR），旨在构造一种奇异点的稳健检出方法[58]。他们的方法主要基于椭球多变量修剪法[59]（MVT）和最小中位方差估计（LMS）法，MVT法主要用于得到一个稳健的离差矩阵，和最小中位方差估计（LMS）法。MVT法主要用于得到一个稳健的离差矩阵，并在此无奇异点干扰的离差矩阵上进行主成分分析以检出 $\boldsymbol{X}$ 矩阵的奇异点，构成此稳健主成分回归法的第一步。接着，采用最小中位方差估计（LMS）法所得的标准化残差来检出量测矢量 $\boldsymbol{y}$ 中的奇异值。

6. 稳健偏最小二乘法

偏最小二乘估计（PLS）为化学计量学中一最常用方法。最近，由Wakeling和MacFie将其稳健化[60]。他们将PLS方法中的对 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 矩阵的荷载矢量 $\boldsymbol{w}$ 和 $\boldsymbol{c}$ 的估计过程稳健化，研究结果表明，对于 $\boldsymbol{Y}$ 矩阵中的奇异点此算法是有效的。这稳健化的代价是有关荷载矢量 $\boldsymbol{w}$ 之间正交关系难以保持。此算法是为对 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 矩阵的奇异点进行独立地消除而设计的，其中的稳健步骤是采用了由Beaton和Tukey提出的双加权法[61]。

（二）诊断方法

诊断方法主要是研究那些对最小二乘估计有较大影响的奇异点的特性，旨在找到一种作图或计算方法来直接将这些奇异点或强影响点检出，以消除它们对回归参数估计的影响。本节将对两类诊断方法，即所谓经典诊断方法和稳健诊断方法，进行较详细的讨论。

1. 经典诊断方法

经典诊断方法主要是基于一些非稳健估计，如最小二乘估计模型和它的残差、均值及协方差矩阵的分析之上。在此我们首先介绍最小二乘估计的投影帽子矩阵 $\boldsymbol{H}$，实因很多诊断方法源出于此。

在化学计量学中称为直接校正的多元线性回归模型可写成如下的矩阵形式

$$\boldsymbol{y}=\boldsymbol{X\alpha}+\boldsymbol{e} \tag{6-65}$$

式中

$$\boldsymbol{X}=\begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1m}\\ x_{21} & x_{22} & x_{23} & \cdots & x_{2m}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ x_{n1} & x_{n2} & x_{n3} & \cdots & x_{nm}\end{pmatrix}$$

帽子矩阵可定义为

$$H=X(X^tX)^{-1}X^t \tag{6-66}$$

这个$(n\times n)$阶矩阵之所以称为帽子矩阵是因为它可直接将量测矢量转换成为最小二乘的估计值，即

$$\hat{y}=Hy=X(X^tX)^{-1}X^t=X\alpha \tag{6-67}$$

容易证明，帽子矩阵为一幂等对称矩阵，即

$$HH=H$$

$$H=H^t$$

而且

$$\mathrm{trace}(H)=m$$

实因

$$\begin{aligned}\mathrm{trace}(H)&=\mathrm{trace}(X(X^tX)^{-1}X^t)\\&=\mathrm{trace}(X^tX(X^tX)^{-1})=\mathrm{trace}(I_m)=m\end{aligned}$$

式中，trace(·)表示矩阵的迹。

帽子矩阵为一幂等对称矩阵的事实还意味着

$$\begin{aligned}h_{ii}=(H)_{ii}=(HH)_{ii}&=\sum(h_{ij}h_{ji})=\sum(h_{ij}h_{ij})=\sum(h_{ij}^2)\\&=h_{ii}^2+\sum_{i\neq j}(h_{ij})\end{aligned} \tag{6-68}$$

式（6-68）说明$0<h_{ii}<1$。

而

$$\partial \hat{y}_i/\partial y_i=h_{ii}$$

这说明h_{ii}可看成是第i个量测值对其预测值的影响程度的一种度量。这样，如果h_{ii}很大（靠近1），即说明变量i对最小二乘估计有很大的影响。一般说来，当h_{ii}大于它的平均值（m/n）的2倍或3倍时，可认为第i个量测值所对应的x_i是一个强影响或称为杠杆点[62~65]。

在统计学和化学计量学中，马氏距离是另一个常用的诊断工具

$$MD_i=(x_i-\bar{x})C^{-1}(x_i-\bar{x})^t \tag{6-69}$$

在这里C是X矩阵的协方差阵，而$\bar{x}$则是$x_i(i=1,2,\cdots,n)$的均值矢量。可以证明：

$$MD_i=(n-1)[(h_{ii}-1)/n] \tag{6-70}$$

从式（6-70）可以看出，马氏距离与帽子矩阵有类似功效，因它本身就是h_{ii}函数。上述两种诊断方法对于单个的奇异点或杠杆点都有一定作用，但是，由于杠杆点具有很强的相互掩蔽效应，以致造成很多假象而掩蔽了真正的奇异点和杠杆点。出现这种现象的理由可能是帽子矩阵和马氏距离都是建立在不稳健的经典协方差矩阵之上。

如前所述，单独使用帽子矩阵和马氏距离不足以找到在回归分析中的全部奇异点和强影响点，另一个原因可能是由于它们都没有用到量测矢量y的信息。为了估价在量测矢量中第i个量测点的影响，似乎有必要在进行回归分析时比较包括第i个量测点或不包括第i个量测点时的回归参数的结果，这就导致了Cook的平方距离[65]：

$$CD_i^2=[(\hat{\alpha}-\hat{\alpha}(i))^tM(\hat{\alpha}-\hat{\alpha}(i))]/c \tag{6-71}$$

在式（6-71）中，$\hat{\alpha}$是α的最小二乘估计，而$\hat{\alpha}(i)$则是α在没包括第i个量测点时对α的最小二乘估计。如果选择$M=X^tX$和$c=ms^2=m[\sum r_i^2/(n-m)]$，由式（6-71）求得的$CD_i^2$很大，这就说明第$i$个量测点对回归参数的估计有很大的影响。Cook的平方距离还可推广到诊断多个奇异点的情况[63]：

$$CD(I)^2=[(\hat{\alpha}-\hat{\alpha}(I))^tM(\hat{\alpha}-\alpha(I))]/c \tag{6-72}$$

式中，I 代表对应于一个量测点子集的标号，别的都与前式中的意义同。然而，怎样选择包括 I 的子集却的确不容易。很可能有些点在单独存在时影响不大，但如果它们协同作用时影响就大了。这样，各种可能的子集构成都必须加以考虑才行，由此导致的计算就会带来实际的困难。可以想象，如果必须计算所有可能的子集集合，即所有的 $C_n^p(p=1,2,\cdots,n/2)$，这恐怕是目前已有计算机都难胜任的大工作量任务。所以，如何有效地诊断多个奇异点共存而不受奇异点的掩蔽效应影响的问题实际上还是一个未完全解决的问题。

2. 稳健诊断方法

为了克服奇异点的掩蔽效应，Rousseeuw 等提出了一个稳健距离[41]：

$$RD_i=(x_i-T(\boldsymbol{x}))\boldsymbol{C}(\boldsymbol{X})^{-1}(x_i-T(\boldsymbol{x}))^{t} \tag{6-73}$$

如果将式（6-73）与式（6-69）进行比较，容易发现，式（6-73）与它的主要差别在于用 $T(\boldsymbol{x})$和 $\boldsymbol{C}(\boldsymbol{X})$分别代替了其中的不稳健的均值矢量$\bar{x}$和协方差阵 $\boldsymbol{C}$。在这里，$T(\boldsymbol{x})$和 $\boldsymbol{C}(\boldsymbol{X})$为所谓的最小体积椭球估计（MVE）[66]。它们可以通过下述迭代加权过程而得到

$$T(\boldsymbol{x})^k=\sum w_i^k\boldsymbol{x}_i/\sum w_i^k \tag{6-74}$$

$$\boldsymbol{C}(\boldsymbol{X})^k=\sum w_i^k(x_i-T(x)^k)^t(x_i-T(x)^k)^t/(\sum w_i^k-1) \tag{6-75}$$

在这里的迭代权重因子 w_i^k 由前一次的稳健距离确定：

$$w_i^k=\begin{cases}1 & \mathrm{RD}_i^k\leqslant c\\ 0 & \text{其他}\end{cases}$$

式中，c 为一截断值，可采用 $\chi^2_{m,0.975}$。

Rousseeuw 等用此稳健距离，结合最小中位方差估计（LMS）以同时检出在 $\boldsymbol{X}$ 和 $\boldsymbol{y}$ 中的奇异点。然而 Cook 和 Hawkins[67]指出这一过程有可能过量检出奇异值、算法参数的很小变化，有时却能显著地改变稳健距离的检出结果。最近，Atkinson 和 Mulira 提出一种新的稳健诊断方法，即所谓的“钟乳石图”来消除多变量奇异点[68]。他们不改变马氏距离中的均值和协方差阵，而是采用逐步扩大计算子集并逐步剔除奇异点的方法来克服多个奇异点的掩蔽效应。

在化学计量学中，Naes 提出了一个为主成分回归（PCR）检出杠杆点或强影响点的诊断方法[69]。Hu 和 Massart 比较研究了几种稳健方法，如单个中位数法，重复中位数法，最小中位方差估计法（LMS）和模糊校正方法的检出奇异点的能力[70]。他们的结论是：在有奇异点存在的情况下，最小二乘估计难于获得正确结果，而稳健方法和模糊校正方法的估计结果一般是可接受的。最近，陈文灿等提出一种基于所谓的“钟乳石图”算法基础的稳健诊断的新算法，可用于多个样本量测矢量的奇异点的同时检出。在有奇异点存在的情况下明显优于最小二乘估计[71]。

（三）降维处理的稳健方法

降维处理方法是化学计量学中的一个常用的分析方法，它可将化学量测中得到的高维数据转变成为易于观察的低维数据，在多元校正与分辨及模式识别中得到了广泛的应用。在这里将对近年来出现的一些稳健降维方法作出简要的介绍。

1. 投影寻踪法

如果一个化学量测的 m 维数据可看成是 m 维空间的一个点，则一组这样的化学量测数据可看成是 m 维空间的一簇点云。降维处理技术的目的就是要找到一种方法，以将这些高维数据点在低维空间中描述出来而尽最做到不改变这些高维点云的基本结构。近年来在多元统计研究中发展起来的投影寻踪法（PP）就是这样一种降维技术。它首先定义一种投影指标，然

后，找到一种最优方法来实现这种投影。不同的投影指标就定义了不同的投影寻踪方法。Huber 对 PP 方法进行了很好的评述[72]。Friedman 和 Stuetzle 则将 PP 技术加以拓广，提出了投影寻踪回归[73]，投影寻踪分类[74]和投影寻踪密度估计[75]。

实际上，化学计量学中常用的主成分分析（PCA）本身就是一种特殊的 PP 过程。如把化学量测矩阵 $\boldsymbol{X}$ 看成是包括 n 个 m 维数据点的点集，即量测矩阵 $\boldsymbol{X}$ 中每一个行矢量 $\{\boldsymbol{x}_i=(x_{i1},x_{i2},x_{i3},\cdots,x_{im}),i=1,2,\cdots,n\}$ 看成是一个数据点，同时记量测矩阵 $\boldsymbol{X}$ 的协方差阵为 $\boldsymbol{V}$，有 $\boldsymbol{V}=\boldsymbol{X}^t\boldsymbol{X}$。用 $\lambda_1,\lambda_2,\cdots,\lambda_m$ 表示协方差阵 $\boldsymbol{V}$ 的特征值，则第一主成分轴 $\boldsymbol{a}_1$ 就是将量测矩阵 $\boldsymbol{X}$ 向其投影所得的方差最大的方向，即

$$\lambda_1=\mathrm{Max}(\boldsymbol{a}_1^t\boldsymbol{V}_{a1})=\mathrm{Max}(\boldsymbol{a}_1^t\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{a}_1)\ \|\boldsymbol{a}_1\|=1 \tag{6-76a}$$

$$\lambda_2=\mathrm{Max}(\boldsymbol{a}_2^t\boldsymbol{V}\boldsymbol{a}_2)=\mathrm{Max}(\boldsymbol{a}_2^t\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{a}_2)\ \|\boldsymbol{a}_2\|=1 \qquad (\boldsymbol{a}_2\perp\boldsymbol{a}_1) \tag{6-76b}$$

$$\lambda_m=\mathrm{Max}(\boldsymbol{a}_m^t\boldsymbol{V}\boldsymbol{a}_m)=\mathrm{Max}(\boldsymbol{a}_m^t\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{a}_m)\ \|\boldsymbol{a}_m\|=1 \qquad (\boldsymbol{a}_m\perp\boldsymbol{a}_1,\cdots,\boldsymbol{a}_{m-1}) \tag{6-76c}$$

实际上，$\{\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_m\}$ 就是协方差阵 $\boldsymbol{V}$ 的对应于特征值 $\lambda_i(i=1,2,\cdots,m)$ 的特征矢量。另一方面，如果协方差阵 $\boldsymbol{V}$ 的主成分已知，则协方差阵 $\boldsymbol{V}$ 可以由主成分重构：

$$\boldsymbol{V}=\sum\lambda_i\boldsymbol{a}_i\boldsymbol{a}_i^t \tag{6-77}$$

从（6-76）与（6-77）式可以看出，经典主成分分析因采用最大方差为其投影指标，故对奇异值很敏感，所以它是不稳健的。这暗示如果选择稳健的投影指标，就有可能得到稳健主成分分析。有趣的是在稳健回归与投影寻踪存在着密切关系[41]，正如 Roosseeuw 评论的那样，用投影寻踪来构造高崩溃点的稳健方法是在人意料之中的。

2. 稳健主成分分析和稳健奇异值分解法

李国英和陈忠链提出了一种基于投影寻踪的稳健主成分分析算法，他们将 Huber 的 M 估计的分散度作为投影指标[76]。谢玉珑等也提出了一个基于投影寻踪和模拟退火技术的稳健主成分分析算法，最小中位矢量作为其投影指标[77]。计算机的数值模拟结果显示出此稳健主成分分析方法对非正态分布数据和奇异值具有很强的抵御能力。同样是基于投影寻踪，Ammann[78]提出了一种新的稳健奇异值分解方法(RSVD)。此方法由两步迭代过程所构成：①对数据矩阵进行最小二乘回归拟合，继而将其向回归超平面进行旋转；②通过稳健的广义 M 估计得到协方差矩阵的特征值和特征矢量从而找到描述该数据阵的稳健协方差阵和期望值。此稳健奇异值分解方法（RSVD）可以成为很多多元分析方法的基础，如稳健多元回归，稳健多元判别分析和主成分分析等。

近年来，在统计学和化学计量学的研究中，稳健方法的研究受到日益重视，并取得了显着成果。其发展的动因主要来自对科学和工业研究的实际数据解析的需要，实因在实际数据中，奇异值和非正态数据经常遇到，而量测数据的正态性却又经常是很多现存多元分析方法的数学假设基础，这就使结果的可靠性受到影响。为克服这种假设与实际存在差异的矛盾，稳健方法就应运产生，面且特别是近年来得到飞速发展。化学计量学家们应跟上这一形势的发展，重视稳健方法的研究和在化学数据解析中的实际应用，特别地，对适合于化学数据特征的新的化学计量学稳健方法的研究应该得到应有的重视。

五、非线性体系的校正方法

在分析化学中，一般都是假设被研究体系为线性加合体系，如在光谱分析中就是要服从 Lambert-Beer 定律，这样就可用前述的多元校正法，即直接校正法、间接校正法和广义标准加人法来进行校正，一般说来，都可得到满意的结果，只要体系确实是线性加合体系。然面，作

为一般分析工作者都知道,Lambert-Beer 定律是有条件的,其中最重要的一个条件就是它有一个浓度适应范围,超出这一浓度适应范围,Lambert-Beer 定律就不成立了。另外,在波谱量测中还经常出现部分非线性的情况,即在有些波长点出现不符合线性加合性。这就向化学工作者提出了新问题,即怎样进行非线性多元校正的问题。对于部分非线性问题,一般可以采用前节所讨论的稳健方法来进行校正,把那些非线性的量测点看成是异常值处理,但是一般要求这样的非线性量测点不超过 50%。在分析化学中还存在着很多的非线性现象,而且,对于这些非线性的体系一般还很难找到适当的数学模型,是不是还存在着一种方法可对一般非线性体系进行多元校正。近年来,对于非线性多元校正成了分析化学计量学研究的一个热点,出现了不少的非线性校正方法,如非线性偏最小二乘法等。这些方法大都是在线性模型里引进非线性项,以补偿体系中的非线性,在此不详细介绍,读者可参阅文献[79],这里主要介绍人工神经网络法。

人工神经网络是从 20 世纪 50 年代末、60 年代初开始的,代表性工作是 Rosenblattd 的感知器(perceptron)和 Widrow 的自适应线性元(adalnei)[80],至今已有 30 余年历史。特别是 20 世纪 80 年代以来,以 Hopfield 等人[81,82]的联想记忆网络模型、制约优化网络模型、Rumelhart 等人[83]的多层前传网络误差反传算法(BP 算法)等为代表的一批优秀工作的出现,掀起了人工神经网络研究的热潮。因人工神经网络具有很强的非线性映照能力而受到化学计量学家们的普遍关注,是目前一个发展很快的化学计量学研究领域。本节只对人工神经网络的基本概念及多层前传网络误差反传算法(BP 算法)作出介绍,以期读者对其有一个初步认识。

【基本思路】

人工神经网络的基本思路是基于人脑细胞(神经元)的工作原理来模拟人类思维方式,以建立模型来进行分类与预测的。图 6-8(a)给出了一个神经元的示意图,从图 6-8 中可以看到,神经元通过神经纤维或突触与别的神经元相联系,用以接受来自别的神经元的信息,并将信息继续传给别的神经元。一个典型的神经元可与 10^3 的别的神经元相连,所以人的大脑是一个相当复杂的系统。实际上,人工神经网络方法只是简单地借用神经元来表示一个计算单元[参见图 6-8(b)],它也可以像神经元一样通过网络与别的计算单元相连,只是被叫成是输入和输出。

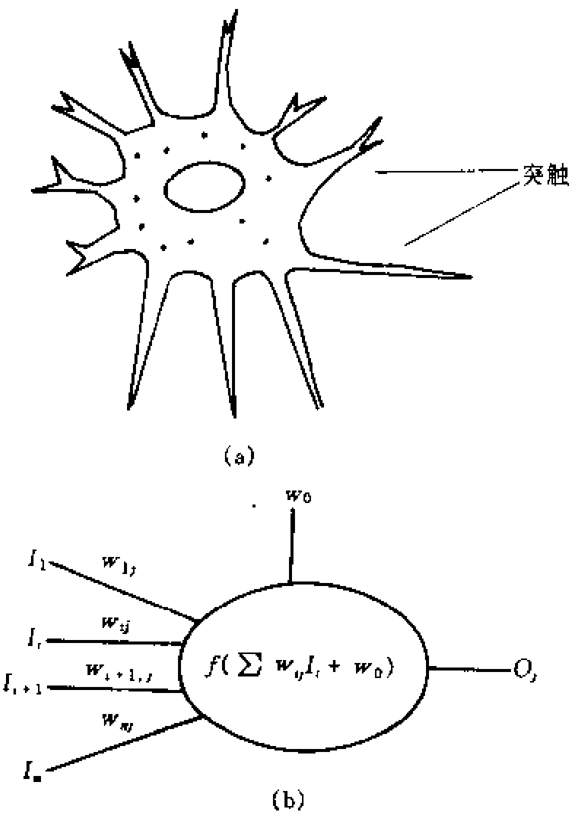


图 6-8 真实神经元示意图(a)与人工神经网络的计算单元(b)

图 6-9 示出了一个典型的基于误差反传算法的三层前传网络,从下至上分别称之为输入层、隐蔽层和输出层。在人工神经网络的计算中,输入层一般表示为 $\boldsymbol{I}$,它的每一个计算单元将代表 $\boldsymbol{I}$ 中的一个元素,它们通过第一层连接权重 w_{ij}^{m-1} 由式(6-78)与隐蔽层的计算单元 y_j^m 相连接,

$$y_j^m = f(\sum w_{ij}^{m-1} I_i + w_0^m) \qquad (6\text{-}78)$$

式中,$f(\cdot)$称为活性函数,一般可以取以下几种形式:

① $f(u) = \mathrm{sgn}(u) = f_h(u)$符号函数,当 u 为正数时,取 1,否则取 -1,此函数用于处理模式

识别时很有用。

② $f(u)=(1+e^{-u})^{-1}=f_s(u)$　s 型非线性函数，其输出值介于 0～1 之间，进行非线性校正或预测时很有用。

③ $f(u)=\tanh(u)$　非线性反正弦函数，其输出值在 +1～−1 之间，对模式识别和非线性回归或预测都可用。

另外，$w_0^{m_1}$ 称为第一层权重偏置点，根据不同的情况可取不同的值。隐蔽层 y^m 以同样的方式，也是通过连接权重来与输出层 $\boldsymbol{O}$ 相连。

所以，整个 3 层前传人工神经网络可以由下式表出：

$$\begin{aligned} O_k &= f(\sum w_{kj}^{m_2} y_j^m + w_0^{m_2}) \\ &= f(\sum w_{kj}^{m_2}(f(\sum w_{ij}^{m_1} I_i + w_0^{m_1})_j) + w_0^{m_2}) \end{aligned} \tag{6-79}$$

式中，$w_{kj}^{m_2}$ 表示第二层的连接权重，而 $w_{ij}^{m_1}$ 表示第一层的连接权重。从式(6-79)可以看出，3 层前传人工神经网络实际是通过一个相当复杂的非线性函数将输入矢量 $\boldsymbol{I}$ 与输出矢量 $\boldsymbol{O}$ 连在一起。实际上，这样的前传神经网络还可以是多于 3 层，亦即 4 层或 5 层的，其结构可顺此类推。从此式还可以看出，前传神经网络的确定实质上就是连接权重的确定，因在此式中，$f(\cdot)$、$\boldsymbol{I}$ 和 $\boldsymbol{O}$ 都是可预先确定的，只要权重确定以后，整个网络也就唯一确定了。

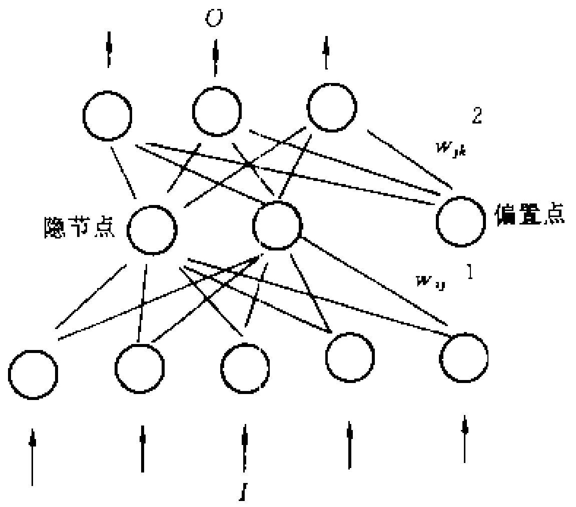


图 6-9　3 层前传人工神经网络示意图

人们对多层前传网络解析能力的认识首先来源于大量的模拟实验和工程实践的感性认识。在应用多层前传网络解决模式识别问题的研究中，Lippmann 通过大量的仿真实验得出结论[84,85]：采用 3 层网络可以形成若干个复杂的决策域；而采用 4 层网络则可以形成任意复杂的决策域。Wieland 和 Leighton 则通过一个具体的例子说明 3 层网络具有把指定空间分解成为若干个子空间的能力[86]。因此，大量的仿真结果很自然地给人们一种启示，多层前传网络可能具有实现任意复杂非线性映射的能力。

为了建立人工神经网络模型，亦即确定连接权重，Rumelhart 等人[87]提出了多层前传网络误差反传算法（BP 算法）。也正是这一算法的提出，使得人工网络的研究又形成了一个热潮。这一算法在模式识别及非线性回归方面得到了非常广泛的应用。所谓误差反传的基本思路就是定义一个误差函数作为训练网络的目标函数，然后采用一种方法根据误差反馈过来的情况以调节网络的连接权重，从而达到优化目的。一般来说，都是采用最小二乘函数来作为目标函数，即使式（6-80）极小化

$$E = 1/2(\sum(O_k - E_k)^2) \tag{6-80}$$

式中，$E_k(k=1,\cdots,L)$ 为期望所得到的输出值、L 为输出矢量的元素个数。

【基本算法】

BP 算法将由以下几步构成：

① 数据预处理。因为一般大都采用由 s 型非线性函数为其活性函数，即 $f(u)=(1+e^{-u})^{-1}$。因其输出值都介于 0～1 之间，所以一般都需要对网络的输出的数值进行预处理，使之同样落入 0～1 之间。注意到这样的预处理不会影响问题的一般性。

② 随机选取网络之间的连接权重 $w_{kj}^{m_2}$ 和 $w_{ij}^{m_1}$ ($j=1,\cdots,n; i=1,\cdots,m; k=1,\cdots,L$)，其中 n 为输入矢量的节点数（或称输入矢量的维数）；m 为隐节点数；L 为输出节点数。一般就用（$-0.5\sim0.5$）的均匀分布的随机函数产生。

③ 设 $\alpha=1, \cdots, A$，重复迭代进行以下步骤，直至收敛。

1）前传计算：按照

$$y_j^m = f(\sum w_{ij}^{m_1} I_i + w_0^{m_1}) \tag{6-81}$$

和

$$\begin{aligned} O_k &= f(\sum w_{kj}^{m_2} y_j^m + w_0^{m_2}) \\ &= f(\sum w_{kj}^{m_2}(f(\sum w_{ij}^{m_1} I_i + w_0^{m_1}))_j) + w_0^{m_2}) \end{aligned} \tag{6-82}$$

来计算 $\boldsymbol{O}$，并与所期望得到的数值比较，计算出误差函数 E。

2）反传调节：计算出的目标函数对连接权重的一阶导数，并以此来调节连接权重 $w_{kj}^{m_2}$ 和 $w_{ij}^{m_2}$。（因在此只介绍最陡梯度法，所以用的是一阶导数，别的优化算法同样是可行的）。

$$\Delta w_{ij} = \rho \delta_i u_j \tag{6-83}$$

式中，ρ 为一步长因子，可在（0～1）之间取值；u 为第 j 节点上的抽象变量，一般视其是在输出层还是隐蔽层，就分别等于 $f(\sum w_{ij}^{m_1} I_i + w_0^{m_1})$ 或者 $f(\sum w_{kj}^{m_2} y_j^m + w_0^{m_2})$；$\delta$ 为梯度因子，对于不同层，它有不同表达式：

$$\delta_i = \begin{cases} (u_i - E_i)(u_i)(1-u_i) & u_i \text{ 为输出节点} \quad (6\text{-}84a) \\ (\sum w_{ij}^{m_1} \delta_k)(u_i)(1-u_i) & u_i \text{ 为隐蔽层节点} \quad (6\text{-}84b) \end{cases}$$

式中，$u(1-u)=u'$。

因

$$u = f(\sum w_{ij}^{m_1} I_i + w_0^{m_1}) = f(x) = 1/(1+e^{-x}) \tag{6-85}$$

所以有

$$\begin{aligned} u' = f''(x) &= (1/(1+e^{-x}))/\mathrm{d}x = -(1+e^{-x})^{-2}(e^{-x})(-1) \\ &= (1/(1+e^{-x}))(1-(1/(1+e^{-x}))) = u(1-u) \end{aligned} \tag{6-86}$$

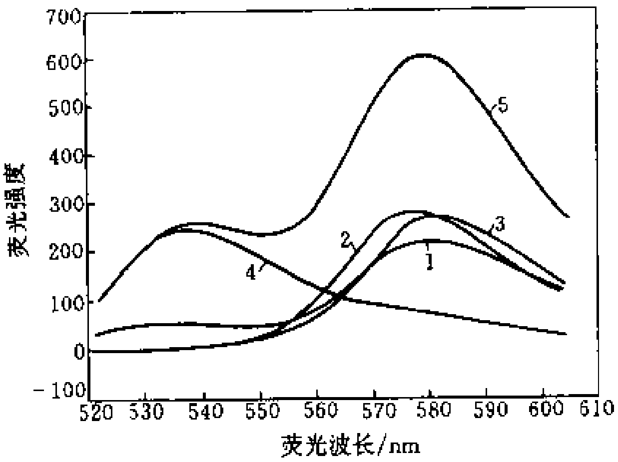


图 6-10 三组分非线性荧光分析体系荧光光谱图

1—三组分混合体系实测荧光强度曲线；2,3,4 分别为罗丹明 B、丁基罗丹明 B、曙红 B 单组分实测荧光强度曲线；5—将曲线 2,3,4 按线性加合计算所得的曲线

式（6-83）中的步长因子 ρ 的取值大小对收敛速度有很大的影响，如果取值太大，则可能引起迭代过程的振荡；反之，若取值太小则会导致权重调节的迭代过程收敛太慢。一般说来，为了加快迭代过程且防止迭代过程的振荡，一般是在式（6-83）中引入一个惯量因子 λ

$$\Delta w_{ij}(\alpha+1) = \rho \delta_i u_j + \lambda \Delta w_{ij}(\alpha) \tag{6-87}$$

以保证迭代收敛速度。

【讨论】

因为人工神经网络具有很强的非线性映照能力，可在非线性多元校正中取得重要作用，是一个值得重视的非线性多元校正的方法。荧光分析中因存在荧光熄灭现象，多组分混合时存在着非线性，而且其数学模型不清楚，从而为荧光分析的多元

校正带来困难。对一个含罗丹明 B、丁基罗丹明 B 和曙红 B 的三组分荧光混合体系进行了调查[88]，所得结果示于图 6-10，从图 6-10 可以看出，此三组分体系的确存在严重非线性，用偏最小二乘（PLS）对其分析的结果示于表 6-5，说明 PLS 不具有很好的预测能力，这是因为 PLS 实质上还是一种基于线性的数学模型，而采用基于误差反传的前向三层的人工神经网络进行分析的结果却是令人满意的，结果示于表 6-6。将人工神经网络用于化学研究的例子可参阅文献[88，89]。

表 6-5　PLS 对试样各组分浓度预测的相对误差（%）

试样编号	罗丹明 B	丁基罗丹明 B	曙红 B	试样编号	罗丹明 B	丁基罗丹明 B	曙红 B
1	4.841	12.809	14.652	17	8.286	−23.076	−28.202
2	1.866	−1.514	−6.683	18	0.660	−9.675	−1.278
3	−7.982	2.611	22.768	19	8.076	−0.221	2.668
4	27.961	31.952	42.056	20	2.866	−11.133	−15.747
5	−0.372	0.839	−4.028	21	8.277	−23.468	−24.575
6	10.816	4.502	3.898	22	−22.460	0.366	4.781
7	0.439	1.580	1.759	23	−17.494	−21.628	−26.986
8	−2.948	10.271	7.699	24	−21.502	−14.432	−16.078
9	4.732	7.281	7.638	25	6.821	9.258	10.737
10	26.259	4.587	18.773	26	2.345	11.987	10.670
11	−58.106	−26.017	−44.797	27	10.335	7.050	14.199
12	0.500	−13.927	−11.580	28	4.557	3.902	3.219
13	−7.042	−0.099	6.747	29	−13.466	6.116	−12.01
14	−24.314	−3.130	−6.833	30	4.315	2.311	14.686
15	−45.161	36.420	29.213	绝对值平均	12.871	10.247	14.058
16	−28.326	−5.238	6.770				

表 6-6　BP-ANN 对试样作为未知集时各组分浓度预测的相对误差①

试样编号	罗丹明 B			丁基罗丹明 B			曙红 B		
	c_0/ μg·ml⁻¹	c_p/ μg·ml⁻¹	e_p/%	c_0/ μg·ml⁻¹	c_p/ μg·ml⁻¹	e_p/%	c_0/ μg·ml⁻¹	c_p/ μg·ml⁻¹	e_p/%
1	2.994	3.030	1.213	1.892	1.832	−3.164	5.971	5.635	−5.621
2	3.368	3.119	−7.392	2.648	2.618	−1.144	5.598	5.450	−2.644
3	1.871	1.787	−4.488	1.892	1.906	0.785	6.344	6.182	−2.552
4	1.871	1.917	2.442	2.648	2.650	0.053	6.717	7.052	4.981
5	2.994	2.931	−2.111	3.405	3.494	2.615	7.090	7.317	3.196
6	3.368	3.518	4.449	3.027	2.957	−2.311	7.464	7.301	−2.175
7	4.117	4.105	−0.287	3.405	3.305	−2.947	9.703	9.926	2.304
8	4.865	4.620	−5.041	4.162	4.115	−1.133	10.822	10.453	−3.416
9	4.491	4.486	−0.119	3.784	3.695	2.346	10.076	9.923	−1.518
10	2.994	3.171	5.902	5.297	5.116	−3.420	9.330	9.501	1.833
11	2.245	2.068	−7.914	1.513	1.603	5.951	3.732	3.634	−2.611
12	4.491	4.433	−1.292	3.027	3.061	1.126	10.076	9.814	−2.601
13	2.620	2.719	3.774	1.892	1.770	−6.446	5.224	5.316	1.752
14	1.871	1.988	6.262	2.270	2.147	−5.436	4.478	4.440	−0.848
15	1.497	1.497	0.050	1.513	1.667	10.18	4.105	4.302	4.814
16	2.245	2.224	0.931	1.892	1.822	−3.695	4.478	4.468	−0.216
17	1.871	2.038	8.899	3.027	3.073	1.534	4.851	5.154	6.252
18	1.492	1.659	10.82	2.648	2.718	2.644	4.478	5.007	11.82
19	4.117	4.512	9.592	4.162	4.124	−0.917	11.196	11.379	1.638

续表

试样编号	罗丹明 B			丁基罗丹明 B			曙红 B		
	c_0/μg·ml^{-1}	c_p/μg·ml^{-1}	e_p/%	c_0/μg·ml^{-1}	c_p/μg·ml^{-1}	e_p/%	c_0/μg·ml^{-1}	c_p/μg·ml^{-1}	e_p/%
20	3.743	3.878	3.630	3.784	3.601	−4.836	11.569	11.375	−1.672
21	2.994	3.214	7.345	5.676	5.217	−8.072	11.942	11.777	−1.380
22	5.614	5.169	−7.919	4.540	4.700	3.518	11.569	10.963	5.237
23	3.368	3.442	2.186	4.162	4.245	1.996	11.196	10.708	−4.357
24	4.491	4.339	−3.386	5.297	5.271	−0.487	11.912	11.731	−1.770
25	3.743	3.508	−6.268	4.162	4.098	−1.525	7.464	6.978	6.499
26	4.491	4.158	−7.424	2.648	2.806	5.943	8.583	8.016	−6.604
27	3.743	3.954	5.643	3.027	2.845	6.004	7.837	7.908	0.905
28	4.117	4.135	0.449	3.784	3.745	−1.013	9.330	9.740	4.403
29	2.620	2.798	6.802	3.405	3.741	9.862	7.464	7.137	−4.373
30	4.491	4.679	4.173	3.784	3.730	−1.425	8.210	9.078	10.57
平均			4.696			3.118			3.685

① c_0 为组分实际浓度；c_p 为组分预测浓度；e_p 为未知集浓度预测相对误差。

尽管人工神经网络的研究已取得令人兴奋的进展，但是，总的说来对它的研究无论在理论还是应用上都还处于初期阶段，还存在很多问题值得探讨，如网络训练速度、过拟合、全局寻优、网络结构与网络能力等问题都存在未知因素，值得进一步研究。有兴趣的读者可参阅文献[90～93]。

六、病态体系和有偏估计方法

在由式（6-1）所表述的直接多元校正的模型中，$x_i(i=1,2,\cdots,n)$ 为纯物质的量测矢量（或为纯光谱，或为其他纯物质谱），注意到在此模型中还隐含着一个假设，即不同的化学物质的量测谱是不同的，用数学语言来说，就是 $x_i(i=1,2,\cdots,n)$ 是线性不相关的，即校正矩阵 $\boldsymbol{X}$ 必须是一个满秩矩阵。可是，在分析化学的实际中往往存在着这样的情况，在某些光谱（如最常用的紫外可见谱）的量测中，有些不同的化学物质（如某些异构体）却具有很相似的光谱，虽然它们的光谱多少有的不同，但却非常相似，这就会使校正矩阵 $\boldsymbol{X}$ 接近亏秩，而使满秩条件很难保证，或者说纯物质谱之间的差别已与量测误差相近，这样的情况在数学上称之为病态。由于校正矩阵 $\boldsymbol{X}$ 接近亏秩，亦即存在着共线性关系，在常用最小二乘的求逆时将带来很大误差，而使由此估计出来的相对浓度 $c_i(i=1,2,\cdots,n)$ 不可靠。为了解决由矩阵 $\boldsymbol{X}$ 接近亏秩而带来的问题，近年来统计学家们提出了所谓的有偏估计方法，其中最著名的是岭回归方法。在本节中，将对岭回归方法作出介绍，并讨论用岭回归可在什么程度上来改善病态体系的相对浓度估计。

与最小二乘估计不同，岭回归是一种有偏统计回归算法，其基本公式为

$$\hat{\boldsymbol{c}}(k)=(\boldsymbol{X}^t\boldsymbol{X}+k\boldsymbol{I})^{-1}\boldsymbol{X}^t\boldsymbol{y} \tag{6-88}$$

式中，$\boldsymbol{X}$ 的意义同前，为直接校正矩阵；$\boldsymbol{y}$ 为混合物的量测矢量；$\hat{\boldsymbol{c}}(k)$为混合物的浓度估计矢量；$\boldsymbol{I}$ 为（$n\times n$）阶单位矩阵；k 为可调参数。由于 k 的引入，岭回归失去了最小二乘回归所具有的无偏估计的特点，故称为有偏估计。

岭回归与经典最小二乘回归本质不同点在于它不是以$(\hat{\boldsymbol{y}}-\boldsymbol{y})^t(\hat{\boldsymbol{y}}-\boldsymbol{y})$，即残差平方和趋于最小为其目标，而是使（$\hat{\boldsymbol{c}}-\boldsymbol{c})^t(\hat{\boldsymbol{c}}-\boldsymbol{c}$），即估计参数趋于最小，而这却正是分析工作的目的。对于最小二乘估计参数的估计准确度，可由估计值均方差标准（MSE）给出：

$$MSE(c)=E\{(\hat{c}-c)^{t}(\hat{c}-c)\}=E\{\|\hat{c}-c\|^{2}\}$$

式中，$E\{\cdot\}$表示数学期望，从 MSE 的推导（$=\sigma^2\mathrm{tr}(X^tX)^{-1}=\sigma^2\mathrm{tr}(Q^tX^tXQ)^{-1}=\sigma^2\sum 1/\lambda_i$）可知，$MSE=\sigma^2\sum 1/\lambda_i$，$\sigma^2$ 其中为量测误差之方差，λ_i 为（X^tX）的第 i 个特征值。这一关系说明，最小二乘估计的均方差由 λ_i 决定，其中特别是由最小的特征值所决定。当组分纯光谱（或称纯物种量测谱）的相似程度，也就是我们常说的重叠程度很严重时，即校正矩阵中存在共线性关系时，校正矩阵协差阵的特征值的最小值就会很小，此时体系处于“病态”情况，由最小二乘估计出的 $\hat{c}$的均方差就会很大。而对于岭回归有

$$MSE(c)=\sigma^2\sum 1/(k+\lambda_i) \tag{6-89}$$

从而使估计值的均方差减小以致准确度提高。具体地说，可期望找到某个 $k>0$，使得 $c(k)$的均方差比 $\hat{c}$的小，即

$$E\{\|\hat{c}(k)-c\|^{2}\}<E\{\|\hat{c}-c\|^{2}\}$$

对于 k 值的确定，一般可采用岭迹图法，即根据式(6-88)，用 $\hat{c}(k)$对 k 直接作图。文献[91]给出了一个简易计算方法。但对于有 MatLab 软件的读者，直接就用式（6-88）计算也很容易。

【讨论】

为验证此法的功效，用它对 3 个不同的分析体系进行了分析，并与经典最小二乘回归及 Kalman 滤波方法进行了比较，发现对于病态体系，岭回归方法的确优于上述两法。这三个分析体系为：① 间甲酚、对甲酚、邻甲酚三元混合体系；② 苯酚、间甲酚、对甲酚、邻甲酚四元混合体系；③ 苯酚、间甲酚、对甲酚、邻甲酚、水杨酸五元混合体系；它们的标准纯物质光谱图示于图 6-11。从图 6-11 可以看到，它们的重叠和相似程度是严重的，体系存在共线性关系，为病态分析体系，用常规最小二乘回归和 Kalman 滤波方法均难取得准确估计结果。表 6-7 列出了岭回归、常规最小二乘回归和 Kalman 滤波方法的比较结果，从此可以看出，岭回归的分析结果明显优于最小二乘回归和 Kalman 滤波方法。当体系的“病态”情况更严重时（五组分混合体系），最小二乘回归和 Kalman 滤波方法皆出现“负”的估计浓度，这说明对于严重病态体系，上述两种方法所得结果的可靠性很差。

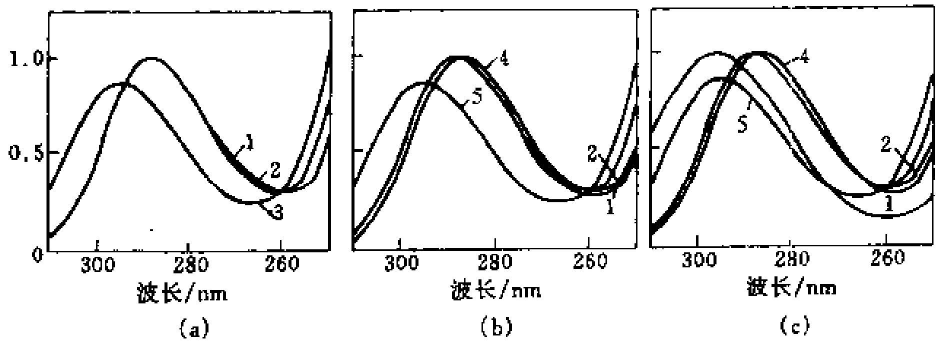


图 6-11　三个标准体系的标准光谱图

(a) 三元体系；(b) 四元体系；(c) 五元体系

1—邻甲酚；2—间甲酚；3—对甲酚；4—苯酚；5—水杨酸

从 3 个混合体系中各取一岭迹图作为实例示于图 6-12。从图可以看出，当 k 从 0 开始变大时，各个浓度估计值就产生显著变化，这等价于只需在校正矩阵的元素上施加很小的微扰，就使最小二乘估计值发生显著变化（注意到当 $k=0$ 时，岭回归就退化成最小二乘回归），说明病态体系的最小二乘估计很不稳定；随着 k 值的增大，岭回归浓度估计值很快趋于平稳，这时

表 6-7　不同方法分析结果的比较

体系①	浓度/$\mu g \cdot ml^{-1}$	最小二乘回归	$MRPE$/%	卡尔曼滤波法	$MRPE$/%	岭回归	$MRPE$/%
[illegible]元体系	A:10.0	10.5	13.2	10.2	7.25	10.1	2.29
	B:10.0	8.08		8.85		9.72	
	C:10.0	11.1		10.5		9.74	
	A:10.0	10.8	7.32	10.9	7.84	10.6	7.12
	B:5.0	4.86		4.93		4.99	
	C:8.0	7.45		7.37		7.27	
	A:4.0	4.4[illegible]	20.3	4.92	18.3	4.25	5.67
	B:2.0	1.29		2.10		1.85	
	C:3.0	3.68		2.66		2.91	
	A:10.0	11.0	9.31	10.0	21.4	10.6	5.19
	B:8.0	8.23		5.53		7.40	
	C:10.0	8.90		12.5		9.82	
四元体系	A:10.0	11.6	32.5	9.69	36.1	10.0	2.5[illegible]
	B:10.0	7.71		4.21		9.84	
	C:10.0	6.25		11.6		9.55	
	D:10.0	14.5		14.0		9.81	
	A:10.0	12.7	34.1	12.7	34.7	9.71	6.63
	B:10.0	8.42		8.22		11.0	
	C:10.0	5.46		5.50		9.23	
	D:10.0	14.0		14.2		10.0	
	A:10.0	11.6	19.6	11.7	25.4	10.7	6.88
	B:6.0	4.33		3.32		6.25	
	C:9.0	7.88		8.26		8.62	
	D:7.0	8.87		9.59		6.24	
	A:9.0	11.4	27.8	11.5	50.26	10.4	8.70
	B:9.0	6.90		2.71		8.35	
	C:8.0	5.59		7.22		7.73	
	D:10.0	13.0		16.0		9.99	
五元体系	A:10.0	13.9	74.9	12.0	62.9	8.59	8.29
	B:10.0	3.11		3.12		9.06	
	C:10.0	3.84		1.60		10.7	
	D:10.0	24.2		20.8		10.1	
	E:10.0	8.53		9.21		9.78	
	A:5.0	5.97	46.2	5.37	32.2	4.44	7.27
	B:5.0	3.24		3.30		4.67	
	C:5.0	2.34		3.69		5.47	
	D:5.0	8.93		7.86		5.09	
	E:5.0	4.58		4.77		4.93	
	A:3.0	4.94	49.1	2.7[illegible]	20.7	2.57	13.0
	B:7.0	4.56		4.87		5.66	
	C:3.0	1.50		3.[illegible]		3.71	
	D:8.0	13.9		9.89		8.87	
	E:8.0	7.28		7.98		8.01	
五元体系	A:15.0	17.4	56.5	16.5	57.1	14.2	14.5
	B:5.0	1.07		1.08		2.86	
	C:0.0	−6.20		−4.16		0.08	
	D:10.0	18.8		17.2		11.7	
	E:9.0	8.11		8.39		9.12	

① A　对甲酚；B—邻甲酚；C—间甲酚；D—苯酚；E—水杨酸；$MRPE=(\sum(c_i-\hat{c}_i)^2/\sum c_i^2)^{1/2}\times 100$。

微扰对解几乎不产生影响，说明岭回归的浓度估计是比较稳定的。在此值得指出的是，有少数情况岭迹图并不完全趋于平稳，此时的 k 值较难确定，这说明对此分析病态体系，岭回归亦难给出满意结果，此时，对这样的体系仍需再用别的有偏估计方法（如广义岭回归、主成分回归、偏最小二乘等）继续证实所估浓度的可靠程度。

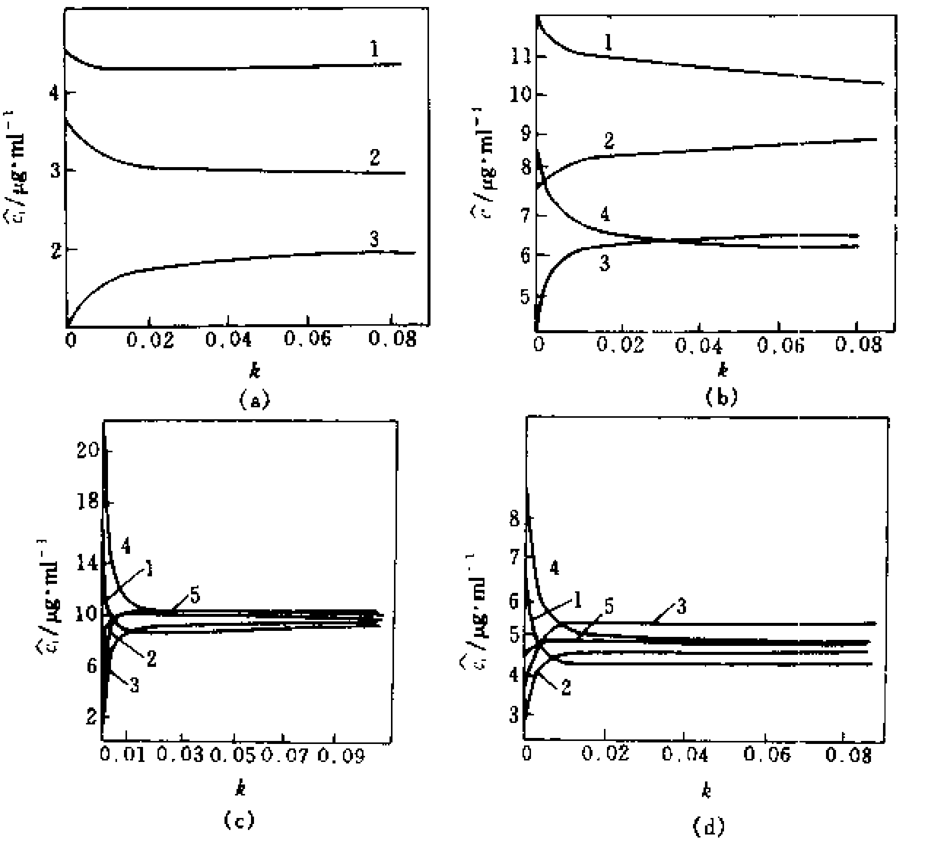


图 6-12　3 个多组分体系的岭迹图

1—对甲酚；2—间甲酚；3—邻甲酚；4—苯酚；5—水杨酸

各组分实际浓度：(a) 1—4μg·ml^{-1}；2—3μg·ml^{-1}；3—2μg·ml^{-1}；(b) 1—10μg·ml^{-1}，2—9μg·ml^{-1}，3—6μg·ml^{-1}，4—7μg·ml^{-1}；(c) 皆为 10μg·ml^{-1}；(d) 皆为 5μg·ml^{-1}

七、多元校正的分析化学品质因数和可靠性分析

分析方法的品质因数，如灵敏度、选择性、检测下限等，在分析化学中是非常重要的概念，借此可对分析结果的可靠性和适应性给出合理估价，亦可作为选择方法和优化实验条件的目标函数。所以，将传统标量校正中习用的有关分析方法的品质因数拓广至多元校正就具有十分重要的意义了。多元校正实质是用矢量分析（多变量）替代（单变量）分析，是传统标量分析的拓广。从前几节的讨论中可以看出，经此拓广，可获得很多传统标量分析所不能得到的重要信息。值得提出的是，多元校正同样可给出传统标量校正中习用的有关分析方法的品质因数[2]。在讨论多元校正的有关分析方法的品质因数之前，有必要先对传统标量校正中习用的有关分析方法的品质因数进行简单回顾。

传统标量校正的模型一般可由式 (6-90) 给出

$$y=cx+e \tag{6-90}$$

式中，y 为量测值；c 为浓度；x 为仪器的灵敏度[95]，或称为响应系数；e 为量测误差，一般

假设为零均等方差的正态分布随机变量，根据 IUPAC 建议，检测下限由式（6-91）定义[96]

$$c=3\sigma/x \tag{6-91}$$

式中，σ 为量测误差的标准差。选择性一般是相对于共存干扰而言，对一含有 n 被测物种体系，如某物种 i 的灵敏度 x_i 为较大正数，其余 $x_j(j=1,2,\cdots,i-1,i+1,\cdots,n)$ 均等于零或为接近噪声的响应，则称该分析方法对物种 i 具有完全选择性，反之，该法对物种 i 的选择性差。

对于多元校正方法，上述品质因数亦可导出，只是表示方法由矢量取代了标量。多元校正的矢量表示模型可由式（6-92）给出。

$$\boldsymbol{y}=c_1\boldsymbol{x}_1+c_2\boldsymbol{x}_2+\cdots+c_n\boldsymbol{x}_n+\boldsymbol{e} \tag{6-92}$$

式中，$\boldsymbol{y}$ 为混合物在 m 个不同分析通道（如光谱则为不同波长或波数）上所测值组成的矢量；$\boldsymbol{x}_i$ 为 i 物种在相应分析通道测得的标准物量测矢量；c_i 为物种 i 在混合物中的浓度；$\boldsymbol{e}$ 为一等方差正态分布的误差矢量。对混合物光谱 $\boldsymbol{y}$，可用构造正交投影矩阵方法来进行解析，先按式（6-93）构成正交投影矩阵 $\boldsymbol{P}_i$。

$$\boldsymbol{P}_i=(\boldsymbol{I}_{m\times m}-\boldsymbol{X}_i\boldsymbol{X}_i^{+}) \tag{6-93}$$

式中，$\boldsymbol{X}_i=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{i-1},\boldsymbol{x}_{i+1},\cdots,\boldsymbol{x}_n)$，即不含 $\boldsymbol{x}_i$ 的校正矩阵，$\boldsymbol{X}_i^{+}$ 为其广义逆，又称 Moore-Penrose逆，在此 $\boldsymbol{X}_i^{+}=(\boldsymbol{X}_i^{t}\boldsymbol{X}_i)^{-1}\boldsymbol{X}_i^{t}$，（有关广义逆和投影阵可参见第十一章的相应部分）。$\boldsymbol{P}_i$ 为一对称幂等矩阵，故有投影阵之称。用投影阵 $\boldsymbol{P}_i$ 作用于混合物量测矢量 $\boldsymbol{y}$ 之上，即将其投影至其正交补空间上，则可把 $\boldsymbol{X}_j(j=1,2,\cdots,i-1,i+1,\cdots,n)$ 的影响扣除，即

$$\boldsymbol{y}_i^{*}=\boldsymbol{P}_i\boldsymbol{y}=\boldsymbol{P}_i(c_1\boldsymbol{x}_1+c_2\boldsymbol{x}_2+\cdots+c_n\boldsymbol{x}_n+\boldsymbol{e})=c_i\boldsymbol{x}_i^{*}+\boldsymbol{e}^{*} \tag{6-94}$$

这是因为投影阵 $\boldsymbol{P}_i$ 具有一优良特性

$$\begin{aligned}\boldsymbol{P}_i\boldsymbol{x}_j&=(\boldsymbol{I}_{m\times m}-\boldsymbol{X}_i\boldsymbol{X}_i^{+})\boldsymbol{x}_j=\boldsymbol{x}_j-\boldsymbol{X}_i\boldsymbol{X}_i^{+}\boldsymbol{x}_j\\&=\boldsymbol{x}_j-\boldsymbol{x}_j=\boldsymbol{0}\quad(j=1,2,\cdots,i-1,i+1,\cdots,n)\end{aligned} \tag{6-95}$$

在此我们利用了广义逆的 $\boldsymbol{X}_i\boldsymbol{X}_i^{+}\boldsymbol{X}_i=\boldsymbol{X}_i$ 的性质。

为对式(6-95)求解 c_i，只需对方程两边左乘 $\boldsymbol{x}_i^{*t}$，可得

$$\boldsymbol{x}_i^{*t}\boldsymbol{y}_i^{*}=\|\boldsymbol{x}_i^{*}\|\cdot\|\boldsymbol{y}_i^{*}\|\cos(\theta) \tag{6-96}$$

式 $\boldsymbol{y}_i^{*}=c_i\boldsymbol{x}_i^{*}+\boldsymbol{e}^{*}$ 中 $\boldsymbol{y}_i^{*}$ 与 $\boldsymbol{x}_i^{*}$ 只相差一误差矢量，故二者之间的夹角 $\theta\approx0$，即 $\cos(\theta)\approx1$，于是有

$$\boldsymbol{x}_i^{*t}\boldsymbol{y}_i^{*}\approx\|\boldsymbol{x}_i^{*}\|\cdot\|\boldsymbol{y}_i^{*}\|\approx c_i\boldsymbol{x}_i^{*t}\boldsymbol{x}_i^{*}+\boldsymbol{x}_i^{*t}\boldsymbol{e}^{*}\approx c_i\|\boldsymbol{x}_i^{*}\|\cdot\|\boldsymbol{x}_i^{*}\|+\boldsymbol{e}^{*} \tag{6-97}$$

对式(6-97)两边同时除以 $\|\boldsymbol{x}_i^{*}\|$ 得

$$\|\boldsymbol{y}_i^{*}\|\approx c_i\|\boldsymbol{x}_i^{*}\|+\boldsymbol{e}^{*}/\|\boldsymbol{x}_i^{*}\|\approx c_i\|\boldsymbol{x}_i^{*}\|+\boldsymbol{e}^{**} \tag{6-98}$$

将式（6-98）与式（6-90）比较，二者有完全一致的表达形式，唯式（6-98）用矢量模替代了式（6-90）中的单变量信号。

1. 灵敏度

从式(6-98)的 $\|\boldsymbol{y}_i^{*}\|\approx c_i\|\boldsymbol{x}_i^{*}\|+\boldsymbol{e}^{**}$ 可知，$\|\boldsymbol{x}_i^{*}\|$ 与式(6-90)中 x 相对应，因此 $\|\boldsymbol{x}_i^{*}\|$ 可称为多元校正中对物种 i 的灵敏度，它表示在单位浓度下多组分共存体系中物种 i 的纯矢量分析讯号大小，即扣除了共存组分影响后的物种 i 的纯分析讯号，其量纲与式（6-90）完全一样。

2. 选择性

对于多组分共存体系，原由式 $y=cx+e$ 表出的标量校正模型应变为

$$y=c_1x_1+c_2x_2+\cdots+c_nx_n+e \tag{6-99}$$

此时仅有一个方程，却存在 n 个未知浓度 $c_i(i=1,2,\cdots,n)$，除非有“完全选择性”，无法求解。而由式（6-98）表出的矢量校正模型无此局限，仍可得标量式解，这说明多元校正模型可提高分析方法的选择性，是其优于传统标量校正模型的基本特征。注意到在多元校正的模型中，可供直接分析用的并非标准量测谱 $\boldsymbol{x}_i$ 本身，而是经投影矩阵 $\boldsymbol{P}_i$ 作用后，与其余共存物标准量测谱的正交部分——投影矢量 $\boldsymbol{x}_i^*$。值得提出的是，有时 $\boldsymbol{x}_i$ 经投影后得 $\boldsymbol{x}_i^*$，其模缩小很多，即 $\|\boldsymbol{x}_i^*\|/\|\boldsymbol{x}_i\|$ 很小。此时用式（6-98）估计浓度，会因 e^{**} 的存在而带来较大误差。这是由于共存物种标准量测谱之间严重重叠所致，通常称为“病态”分析体系，如仍采用最小二乘回归，Kalman 滤波等一般多元校正方法，结果将很不可靠[94]。因此，$\|\boldsymbol{x}_i^*\|/\|\boldsymbol{x}_i\|$ 可视为组分 i 与其他共存组分的标准量测谱之间重叠程度的度量。Lorber[2]建议将其作为组分 i 的选择性指标。然而，作者认为该指标只定性地反映了组分量测谱之重叠趋势，且其可在（0，1）区间变动，对确定某组分是否具有定量分析所需选择性未给出具体指标，很难应用于实际。从式（6-98）可知，多元校正分析的单组分模型，除该被测组分的贡献外只与 e^{**} 有关。也就是说，对于多元校正模型，其选择性可参照单组分模型的定量测定限来考虑，只是应注意到必须用投影矢量 $\boldsymbol{x}_i^*$ 的模，即所谓标准谱的纯分析信号 $\|\boldsymbol{x}_i^*\|$ 来考虑，实因标准谱的纯分析信号 $\|\boldsymbol{x}_i^*\|$ 已经扣除了共存诸组分的共存影响。根据 Currie[97]提出的“定量测定限”之定义，当信噪比大于 10 时，则该组分的分析信号能用于可靠定量测定。综上述讨论，建议将多元校正的组分选择性定义如下：

$$\mathrm{SEL}_i=\|\boldsymbol{x}_i^*\|/e^{**}=\|\boldsymbol{x}_i^*\|/(\boldsymbol{r}\|\boldsymbol{x}_i\|)=(\|\boldsymbol{x}_i^*\|/\|\boldsymbol{x}_i\|)/\boldsymbol{r} \tag{6-100}$$

由式（6-100）定义的多元校正模型的选择性指标表面是一个标准谱的信噪比。但是，实际上它与一般信噪比不同。不同之点在于它是建立在由式（6-98）定义的单组分模型之上的。这是一个经数学变换扣除了共存诸组分的影响后的单组分模型，所以实际上反映了共存诸组分对组分 i 的定量测定的影响。另外，式 $\mathrm{SEL}_i=(\|\boldsymbol{x}_i^*\|/\|\boldsymbol{x}_i\|)/\boldsymbol{r}$ 的 $\boldsymbol{r}$ 为仪器量测的相对误差，这是考虑到分析工作者一般都知道一些仪器的量测相对误差，将其引入可方便选择性指标的运算。对于紫外可见光谱，仪器的标准谱的量测相对误差一般可设为 2%[98]。

根据 Currie 标准，当 $\mathrm{SEL}_i\geqslant 10$ 时，则该物种能由一般多元校正方法可靠定量测定，具有同时测定所需的选择性；当 $3\leqslant\mathrm{SEL}_i\leqslant 10$ 时，因此为定性纯度保证范围，其选择性只满足半定量测定要求；当 $\mathrm{SEL}_i\leqslant 3$ 时，则说明该物种经扣除共存组分影响后的纯分析信号如果小于误差相差的 3 倍，有被误差淹没的可能，此时该组分的选择性不能满足定量分析要求。由式（6-100）定义的组分选择性可方便地对待分析混合试样在解析前就进行选择性估价，只需用投影阵 $\boldsymbol{P}_i$ 对 $\boldsymbol{x}_i$ 进行投影运算而得到 x_i^*，由此可计算出 $\|\boldsymbol{x}_i^*\|/\|\boldsymbol{x}_i\|$，继与仪器的量测相对误差一比较，即可知该体系能否准确定量了。

3. 检测下限

对应式 $y=cx+e$，可定义多元校正的组分检测下限：

$$LOD_i=3\sigma/SEN_i=3\sigma/\|x_i^*\| \tag{6-101}$$

式中，σ 为量测误差标准差，建议使用多个在相同条件所得估计值的均值较为可靠。如 σ 已知，则用实际经验估计的更为合适。

4. 准确度

根据误差传递理论，结果的相对误差，$\Delta c_{i,\mathrm{total}}/c_i$，来自两个方面，即混合物量测相对误差

$\Delta c_{i,y}/c_i$ 和标准物量测误差 $\Delta c_{i,x}/c_i$，可由式（6-102）表出：

$$\Delta c_{i,\text{total}}/c_i = \Delta c_{i,y}/c_i + \Delta c_{i,x}/c_i \tag{6-102}$$

式中，c_i 组分 i 标准量测谱所含浓度。在多组分共存时，在混合物量测中对于组分 i 的信噪比应为混合量测谱中的该组分的纯分析信号 $\| y_i^* \|$ 与噪声标准差 σ 的比值。从式（6-97）可知，

$$c_i \approx x_i^{*t} y_i^* / (\| x_i^* \|)^2 \tag{6-103}$$

代入式（6-98）可得

$$\begin{aligned} \| y_i^* \| &\approx [x_i^{*t} y_i^* / (\| x_i^* \|)^2] \| x_i^* \| + e^{**} \\ &\approx x_i^{*t} y_i^* / \| x_i^* \| \end{aligned} \tag{6-104}$$

即

$$\| y_i^* \| / \sigma = x_i^{*t} y_i^* / (\| x_i^* \| \sigma) \tag{6-105}$$

因 $\Delta c_{i,y}/c_i$ 与在混合物量测中对于组分 i 的信噪比呈倒数关系，则有

$$\begin{aligned} \Delta c_{i,y}/c_i &= \sigma \| x_i^* \| / (x_i^{*t} y_i^*) = \sigma \| x_i^* \| \| y \| / [\| y \| (x_i^{*t} y_i^*)] \\ &= [\| x_i^* \| \| y \| / (x_i^{*t} y_i^*)](\sigma / \| y \|) = \kappa_{i,y}(\sigma / \| y \|) \end{aligned} \tag{6-106}$$

式中，$\kappa_{i,y}$称为混合物量测谱中物种 i 的误差传递数。对于标准物量测谱物种 i 的误差传递数可完全仿此进行，只需将 y_i^* 换成 x_i 即可

$$\begin{aligned} \Delta c_{i,x}/c_i &= \sigma \| x_i^* \| \| y \| / [\| y \| (x_i^{*t} x_i)] \\ &= [\| x_i^* \| \| y \| / (x_i^{*t} x_i)](\sigma / \| y \|) = \kappa_{i,x}(\sigma / \| y \|) \end{aligned} \tag{6-107}$$

于是式(6-102)可写成

$$\begin{aligned} \Delta c_{i,\text{total}}/c_i &= \Delta c_{i,y}/c_i + \Delta c_{i,x}/c_i = \kappa_{i,y}(\sigma / \| y \|) + \kappa_{i,x}(\sigma / \| y \|) \\ &= (\kappa_{i,y} + \kappa_{i,x})(\sigma / \| y \|) = \kappa_{i,\text{total}}(\sigma / \| y \|) \end{aligned} \tag{6-108}$$

在此只假设了标准量测谱与混合物量测谱的量测误差标准差 σ 相等。$\kappa_{i,\text{total}}$称为组分 i 的误差传递数，其大小将说明分析结果的误差大小。

第三节　灰色分析体系的多元校正方法

灰色分析体系的基本特征为：已知待测物存在于待分析的样本中，但是否存在别的未知干扰却不清楚，分析目的是在未知干扰的存在下，直接对感兴趣的待测物进行定量分析。此类分析体系可以说是分析化学家碰得最多的一类体系，因其定性组成只部分已知，介于白色与黑色分析体系之间，故将其称为灰色分析体系。

灰色分析体系的校正算法将以矢量校正方法和矩阵校正方法为其线索，逐步展开。从理论上说，如没有特殊条件，基于矢量校正的方法，如本书将要介绍的迭代目标转换因子分析法（ITTFA），自适应 Kalman 滤波法（AKF），局部曲线拟合法（LCF）等，一般只能给出可能解，虽其实用性受到一定限制，但对一些情况仍有一定参考价值，而且，因其对以后的进一步研究具有启发作用，故本书仍给出了详细的介绍。矩阵校正方法是分析化学计量学研究的难点和热点，并在近年来取得了较大成果。其中著名的有广义秩消失因子分析法（GRAFA），残差双线性分解法（RBL）和约束背景双线性分解法（CBBL）。从纯数学的角度看，这些方法一般皆可给出有物理意义的唯一解，可在实际分析化学的应用中，还存在着一些具体困难值得进一步研究。

一、矢量校正方法

矢量校正方法是用于矢量类型数据模型的一类算法。它们只适应可产生矢量数据的一类仪器，常见的有紫外可见光谱，红外光谱，拉曼光谱，质谱及某些电化学谱，其基本数学模型可由下式给出

$$y=\sum c_i x_i+r+e=Xc+r+e \tag{6-109}$$

式中，y 是混合物的量测矢量，一般假设在 k 个分析通道（可以是波长，波数等）上量测所得；$x_i(i=1,2,\cdots,n)$ 是待测物的纯谱矢量；$c_i(i=1,2,\cdots,n)$ 为待测物的浓度；r 为干扰矢量；e 是量测误差，仍假设为服从正态分布的零均等方差且不相关的白噪声误差。一般说来，有 $k>n$ 的要求，否则无法求解。另外，值得提出的是，式（6-109）的 r 可以是几个干扰物的线性组合，即

$$r=\sum c_j x_j \tag{6-110}$$

式中，$x_j(j=1,2,\cdots,m)$ 为干扰物的纯物种标准谱，$c_j(j=1,2,\cdots,m)$ 为其相应的干扰物的浓度。值得注意的是，从由式（6-109）给出的模型可以看出，不管干扰物是由多少物种构成的，它总被看成为一个背景矢量，这样，可简化我们的模型，但是，这也给求有物理意义的唯一解带来了困难。这在以后的讨论中将逐步展开。

（一）投影算法和多元校正模型的检验

【基本思路】

因灰色分析体系的基本特征是已知待测物存在于待分析的样本，但是否存在别的未知干扰不清楚，故首先确定该分析体系是否确实存在未知干扰就十分重要了。可以说，这是解析灰色分析体系的第一步。如果该体系经检验确实存在干扰物，此时，如用第三章中讨论过的白色分析体系的解析方法，如多元线性回归，Kalman 滤波，偏最小二乘等，将不能获得准确解，必须采用可排除干扰的新算法。投影算法就是一种这样的多元校正模型的检验方法，其基本思路是：如混合样品中确实只含有待测的几种物种，则混合物光谱矢量必可由这几种已知的待测物光谱矢量线性表出，换言之，它必然落在由这几种标准物纯光谱矢量为基矢量所构成的线性空间之中，只需用这几个纯物种光谱组成一正交投影矩阵，对其进行投影运算，即可得一零矢量，也就是说，其正交补空间为一零矢量空间。反之，如混合样品中含有未知干扰物，则其正交补空间不为零，这就构成了该法的数学基础。

【正交投影矩阵的构造及其性质】

要对由（6-109）式表示的灰色体系数学模型进行检验，关键是要找到一个正交投影矩阵来对混合物量测矢量进行投影运算。在构造正交投影矩阵之前，有必要对投影矩阵给出必要的介绍。所谓投影矩阵，是一类对称幂等阵（参见第十一章）。对于由（6-109）式表出的校正模型，作为检验用的正交投影校正可由式（6-111）给出。

$$P=(I-XX^{+}) \tag{6-111}$$

式中，I 为 $(k\times k)$ 阶单位阵，$X=(x_1,x_2,\cdots,x_n)$，即由已知待测物标准谱为列矢量所构成的矩阵，而 X^{+} 则为其广义逆，亦称 Moore-Penrose 逆，一般就是最小二乘逆，即

$$X^{+}=(X^{t}X)^{-1}X^{t} \tag{6-112}$$

根据广义逆的定义（参见第十一章），易知 P 确实为一幂等对称阵，

$$P^{t}=(I-XX^{+})^{t}=I^{t}-(XX^{+})^{t}=I-XX^{+}=P \quad \text{（对称）} \tag{6-113a}$$

$$PP=(I-XX^{+})(I-XX^{+})=I-2XX^{+}-XX^{+}XX^{+}=I-XX^{+}=P \quad (\text{幂等})$$

(6-113b)

如果将由式(6-113)定义的正交投影阵作用在由式(6-109)给出的混合物量测矢量上，则有

$$\begin{aligned} y^{*}=Py&=(I-XX^{+})(Xc+r+e)=Xc-XX^{+}Xc+(I-XX^{+})r+(I-XX^{+})e \\ &=(I-XX^{+})r+(I-XX^{+})e=r^{*}+e^{*} \end{aligned} \tag{6-114}$$

从式(6-114)的推导过程中可以看出，如果式(6-114)给出的混合物量测矢量中没有背景干扰矢量那一项，可知经这样的投影运算后，就只剩下误差项，在此 e^{*} 应仍为一具有零均等方差的随机矢量，反之，若体系中含有未知干扰，则除此误差项外，还有一项 r^{*}，除非 r 可由 $X=(x_1, x_2, \cdots, x_n)$ 线性表出。因有 r^{*} 的存在，y^{*} 就不是一零均等方差的误差矢量了。

【统计检验】

根据以上分析，可建立如下的模型检验方法，首先，用 t 分布检验 y^{*} 是否为一零均的随机矢量，即原假设 $H_0: E(y_i)=0$，令统计量 T 为

$$T=(E(y_i)-0)\times\sqrt{k-1}/S_k$$

在这里，$S_k=\sqrt{\sum[y_i-E(y_i)]^2/k}$；$E(y_i)=\sum y_i/k$。如 $|T|>t_{\alpha/2}$，则否定假设 H_0。$t_{\alpha/2}$ 可由 t 分布表查得。

仅对 $y^{*}=(y_1, y_2, \cdots, y_k)^{t}$ 进行均值检验尚不充分，还需检验 y^{*} 的模，即所谓二范数 $\|y^{*}\|$，是否显著大于量测误差，光度分析的量测误差一般在 1%～2%之间，故可假设其方差由下式估计：

$$\sigma^2=0.01\times\|y\|^2 \tag{6-115}$$

如 $\|y^{*}\|^2/\sigma^2=\|y^{*}\|^2/(0.01\times\|y\|^2)>3$，

即此时的信噪比大于 3，根据分析化学中的检测下限定义[99]，此时，可认为体系中存在未知干扰物，不能认为 y^{*} 是一随机误差矢量。

综上所述，如果

$$|T|>t_{\alpha/2} \quad 且 \quad \|y^{*}\|^2/\sigma^2>3,$$

则可认为校正模型不正确，体系确实存在有未知干扰物，此时不能直接使用白色分析体系的校正方法来进行校正，必须先排除背景干扰。表 6-8 与表 6-9 给出了一些具体实例[100]。

表 6-8　校正模型的检验结果

编号	体系实际存在的物种	校正模型包含的物种	t 检验结果	信噪比	检验结果
1	Cr(Ⅲ)，Ni(Ⅱ)	Cr(Ⅲ)	1.901	28.6	存在干扰
2	Cr(Ⅲ)，Ni(Ⅱ)	Cr(Ⅲ)，Ni(Ⅱ)	−0.0168	0.38	模型正确
3	Cr(Ⅲ)，Ni(Ⅱ)，Co(Ⅱ)	Cr(Ⅲ)，Ni(Ⅱ)	1.883	15.4	存在干扰
4	Cr(Ⅲ)，Ni(Ⅱ)，Co(Ⅱ)	Cr(Ⅲ)，Ni(Ⅱ)，Co(Ⅱ)	0.347	0.754	模型正确
5	蒽		29.3	59.4	存在干扰
6	蒽	蒽	0.964	3.12	模型正确
7	苯并芘、荧蒽	苯并芘、荧蒽	10.48	51.18	存在干扰
8	苯并芘、荧蒽	苯并芘、荧蒽	0.765	0.968	模型正确
9	氨基比林、安替比林、巴比妥	氨基比林、安替比林	0.436	4.79	存在干扰
10	氨基比林、安替比林、巴比妥	氨基比林、安替比林、巴比妥	0.029	0.462	模型正确

表 6-9 计算结果比较

编号①	实际浓度	未经模型检验的分析结果	模型校正后的分析结果
1	Cr(Ⅲ),3.20mg/ml	Cr(Ⅲ),7.46mg·ml	Cr(Ⅲ),3.30mg/ml
	Ni(Ⅱ),2.20mg/ml		Ni(Ⅱ),2.21mg/ml
3	Cr(Ⅲ),6.40mg/ml	Cr(Ⅲ),10.3mg ml	Cr(Ⅲ),6.40mg/ml
	Ni(Ⅱ),2.20mg/ml	Ni(Ⅱ),1.70mg ml	Ni(Ⅱ),2.21mg/ml
	Co(Ⅱ),1.75mg/ml		Co(Ⅱ),1.77mg/ml
7	苯并芘,0.7mg·L^{-1}		苯并芘,0.675mg·L^{-1}
	荧蒽,0.10mg·L^{-1}	荧蒽,0.567mg·L^{-1}	荧蒽,0.107mg·L^{-1}
9	氨基比林,10.0mg·L^{-1}	氨基比林,16.4mg·L^{-1}	氨基比林,10.0mg·L^{-1}
	巴比妥,1.80mg·L^{-1}	巴比妥,1.41mg·L^{-1}	巴比妥,1.79mg·L^{-1}
	安替比林,4.0mg·L^{-1}		安替比林,3.91mg·L^{-1}

① 此表的编号同表 6-8。

从表 6-8 与表 6-9 的结果可以看出，如不经校正模型检验，对于校正模型不确定的分析体系，直接使用常规的白色分析体系的解析方法，如最小二乘回归，进行多组分同时测定，所得结果是不可靠的。值得特别指出的是，对于校正模型的检验，只用 t 检验是不够的，这从表 6-8 中的第 9 号样本所得结果就可以看出。

关于排除干扰可分为两种情况：一种是已知可能存在干扰物种的大致范围，此时可分别记录各可能干扰物种的标准量测谱，并同时对其进行投影运算，继用检索方式以确定干扰物的种类，然后只需将其代入式 (6-109)，继用白色分析体系的解析方法，立即可求出待测物的浓度来，有关例子可参见文献[100]；另一种情况是有关未知干扰物的信息一无所知，这种情况的处理比较复杂，下面即将对这种情况进行详细讨论。

（二）标准加入迭代目标转换因子分析法

目标转换因子分析法曾用于含已知干扰组分的混合物同时测定[101]，但此法很难用于含未知干扰物的多组分分析体系。实因目标转换需设定实验光谱矢量。在组分干扰未知的情况下，实验矢量无从选择；另一方面，因子分析法要求 l 个各组分浓度不同的混合试样以构成数据矩阵。且 l 应大于待测组分数目 n，而分析工作中常只有一份含未知干扰的多组分混合试样，除非借助另外的分离手段。

【基本思路】

在这里将要介绍的标准加入迭代目标转换因子分析法[102]，其主要思路是将标准加入，矩阵投影运算及迭代目标转换等技术结合起来，构成一个新算法，先完成未知干扰背景的检出，从而使其转化成为简单的白色分析体系，继用第三章的直接校正方法即可进行单个或多个待测组分的定量分析测定。

对于由式 (6-109) 表出的灰色分析体系的校正模型，借前一节已讨论过的广义标准加入法，分别对混合试样加入标准待测组分，从面得到一组标准加入系列溶液的量测矢量 $y^i (i=1, 2, \cdots, l-1)$，构成一量测矩阵：$Y=[y^0, y^1, \cdots, y^{l-1}]$，注意，在这里要求 $l>n$，而且，y^0 就是由式 (6-109) 表出的原始量测矢量。继对矩阵 Y 进行主成分分解（奇异值分解）可得

$$Y=USV^t=UT^t \tag{6-116}$$

式中，U 和 V 分别为两个正交标准化矩阵，S 为 $(l \times l)$ 阶对角矩阵。

显然，如果体系中存在未知干扰背景，则 Y 矩阵的秩为 $(n+1)$，否则，其秩为 n。因本节只讨论体系存在未知干扰背景的情况，故需取 $(n+1)$ 个对应于较大特征值（即 S 矩阵中的对角元素之值）的特征矢量来构成主因子阵 $\bar{U}$，因此

$$Y = XC^{t} = URR^{-1}T^{t} \tag{6-117}$$

在式（6-117）中，$X=[x^1, x^2, \cdots, x^n, r]$，即由已知待测物标准光谱为列矢量加上未知背景干扰光谱矢量所构成的矩阵，另外，有

$$C^{t} = C_0^{t} + \Delta C^{t}$$

其中

$$C_0^{t} = \begin{bmatrix} c_1 & c_1 & \cdots & c_1 \\ c_2 & c_2 & \cdots & c_2 \\ \vdots & \vdots & & \vdots \\ c_n & c_n & \cdots & c_n \\ 1 & 1 & \cdots & 1 \end{bmatrix} \tag{6-118}$$

$$\Delta C^{t} = \begin{bmatrix} 0 & \Delta c_{11} & \Delta c_{21} & \cdots & \Delta c_{l-1,1} \\ 0 & \Delta c_{12} & \Delta c_{22} & \cdots & \Delta c_{l-1,2} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & \Delta c_{1n} & \Delta c_{2n} & \cdots & \Delta c_{l-1,n} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{6-119}$$

在此 Δc_{ij} 表示 i 次对第 j 个待测组分加入的浓度，而且在加入时使背景干扰的浓度实际保持不变，在实际标准加入时，为保持这一条件，可采用加入小体积的较浓标准溶液或采用稀释至相同体积的方法。在式（6-117）中，矩阵 R 称为转换矩阵，它将抽象波谱因子矩阵 U（或称得分矩阵）和抽象浓度因子矩阵 T^{t} 转换成为有物理意义的真实标准量测谱矩阵 X 和真实浓度矩阵 C^{t}。

根据因子分析目标转换法，采用已知的 n 个待测物标准量测矢量，即可求出 R 矩阵的 n 列 $r_i(i=1,2,\cdots,n)$ 来，这是因为 $XC^{t}=UT^{t}=URR^{-1}T^{t}$，亦即

$$X = UR \tag{6-120}$$

实际上，如果将式（6-120）写成矢量式，则有

$$x_i = Ur_i \quad (i=1,2,\cdots,n) \tag{6-121}$$

因为式（6-121）中，x_i 为已知矢量，用最小二乘法解此方程，立即可得

$$r_i = (U^{t}U)^{-1}U^{t}x_i \quad (i=1,2,\cdots,n) \tag{6-122}$$

式中，U 为一正交矩阵，所以 $(U^{t}U)^{-1}$ 的求逆很容易，实际上就是对角特征值矩阵的倒数阵。利用式（6-122）可以求得 R 矩阵的 n 列。但是，因对背景干扰量测矢量事先无任何已知信息，只可采用迭代法来求解。

【基本算法】

（1）迭代初始矢量的选择　迭代初始矢量的选择将决定迭代能否收敛至正确值，这就要求迭代初始矢量含有背景干扰矢量的基本信息，而其他待测组分量测矢量的影响则应尽量扣除，迭代结果方可朝背景量测矢量收敛，这是解决问题的关键所在。在这里我们采用待测组分的标准量测矢量构成投影矩阵

$$P = (I - XX^{+}) \tag{6-123}$$

式中，I 为 $(k\times k)$ 阶单位阵，$X=(x_1, x_2, \cdots, x_n)$，即由已知待测物标准谱为列矢量所构成的矩阵；$X^{+}$ 则为其广义逆，亦称 Moore-Penrose 逆，一般就是最小二乘逆，即

$$X^{+} = (XX)^{-1}X^{t} \tag{6-124}$$

用此矩阵 P 对混合物量测矢量 y^0 进行投影运算，所得的投影矢量 y^{0*} 就具有上述所需性质。而且，投影矢量 y^{0*} 与所有待测量测矢量都正交，即

$$y^{0*t}x_i=0 \quad (i=1,2,\cdots,n) \tag{6-125}$$

这是因为

$$\begin{aligned} y^{0*t}x_i &= y^{0*t}x_i = y^{0t}P^tPx^i = y^{0t}P^t(I-XX^-)x_i \\ &= y^{0t}P^t(x_i-XX^+x_i) = y^{0t}P^t(0)=0 \quad (i=1,2,\cdots,n) \end{aligned}$$

在上述推导中，利用了投影矩阵的对称幂等和广义逆矩阵的 $XX^+X=X$，亦即 $XX^+x_i=x_i$ 的性质。正是由于投影矢量 y^{0*} 与所有待测量测矢量都是正交，所以，可以说，这样选出来的初始迭代矢量 y^{0*} 已将各已知待测组分对混合物量测矢量的贡献全部扣除。

(2) 迭代过程　用上述方法得到初始迭代矢量 y^{0*} 后，将其作为背景干扰矢量的初始估计 f^0，并用其取代式 $x_i=Ur_i$中的 x_i，由式 (6-122) 求出 r_i 后，继用式 $x_i=Ur_i$求新的背景干扰矢量 f^1，仿此可得 f^2，如此循环迭代，直至收敛，即 $\|f^{d+1}-f^d\|$ 小于某一给定正值时，终止迭代，此时所得的最终矢量则为可能的背景干扰矢量。此法求出的背景干扰只是一种可能解，当背景干扰谱与待测物种的量测矢量的重叠度不大时，用此法求得的背景干扰矢量十分靠近真实的背景干扰，但是当背景干扰谱与待测物种的量测矢量的重叠度很大时，一般很难得到真实解。

(三) 自适应 Kalman 滤波法

在第三章中对 Kalman 滤波法用于白色分析体系的解析作了详细地介绍，特别地对 Kalman滤波的新息序列可用于模型校验作出了讨论。实际上就是利用新息序列的特征，自适应 Kalman 滤波法可用于求解灰色分析体系问题[103,104]。

常规 Kalman 滤波算法方程可总结列于表 6 10。

表 6-10　常规 Kalman 滤波算法方程

状态估计模型	$c(k)=Ic(k-1)$	(6-126)
量测模型	$y(k)=x(k)^tc(k)+r(k)$	(6 127)
Kalman 增益	$k(k)=P(k)x(k)[x(k-1)^tP(k)x(k-1)+r(k)]^{-1}$	(6-128)
误差协方阵估计	$P(k)=[I-g(k-1)x(k)^t]P(k-1)[I-g(k-1)x(k)^t]^t+g(k-1)r(k)g(k-1)^t$	(6-129)
状态估计递推	$c(k)=c(k-1)+k(k)[y(k)-x(k)^tc(k)]$	(6-130)

在表 6-10 中，$c(k)$为量测体系在第 k 个量测点上的浓度估计矢量；$x(k)$表示在第 k 个量测点上的系数矢量，即通常所说的敏感系数，可以是摩尔吸光系数等；$y(k)$为在第 k 个量测点上混合物的量测值；$r(k)$为在第 k 个量测点上的量测误差；$k(k)$则称为第 k 个量测点上的Kalman 增益矢量；$P(k)$为在第 k 个量测点上的量测协方差阵的估计。新息序列的定义是：$v(k)=y(k)-x(k)^tc(k)$，实际上就是第 k 个量测点上的估计残差，这个信息对调节递归的动态模型很有用，故在 Kalman 算法中占有十分重要的地位。自适应 Kalman 滤波就是建立在这一概念之上。

首先，当校正模型不正确时，亦即当混合物测矢量中含有未知干扰时，如果校正模型正确，新息序列应该为一具有零均不相关的白噪声系列，实因 $v(k)$就是第 k 个量测点上的估计残差。估计残差是一个很有用的指标，如果估计残差比较大且具有一定的相关性，就说明用于校正的模型存在问题，被估计的参数当然也不正确。因此，正如第二节所讨论的那样，新息序列是一个检验校正模型的很好指标。在此，以一个例子来说明。图 6-13 (a) 示出了一个由 3 个 Guass 峰模三个组分 (a，b，c) 构成的混合物光谱，如果只用两个组分 (a，b) 来构

成校正模型，所得新息序列如图 6-13（b）所示。在第三个组分（c）还没有出现之前，Kalman 滤波的新息序列明显是具有白噪声性质的，这说明校正模型在已被滤波的这一段是正确的。然而，一旦第三组分开始有吸收时，新息序列开始变大且具相关性质，这说明校正模型已不正确了，在这样错误校正模型下估计的浓度亦将不准确了。从上述结果可以看出，因为Kalman 滤波是递归进行的，当校正模型正确时，新息序列给出模型正确的信息，如果模型发生错误，新息序列马上发生变化，给出模型错误的信息。这就提示我们，只要能利用新息序列所提供的这些信息，就有可能将 Kalman 滤波技术用于灰色分析体系的解析，这就是将要讨论的自适应 Kalman 滤波方法。

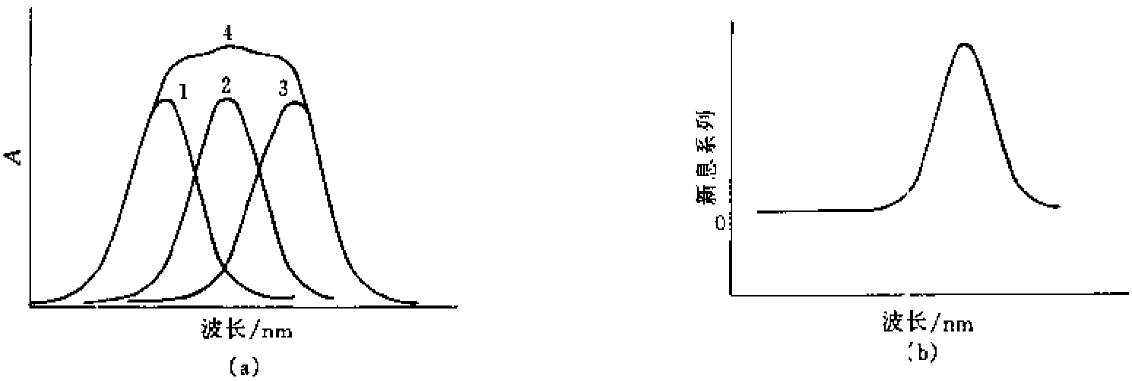


图 6-13　自适应 Kalman 滤波解析灰色体系原理示意图

注：(a) 三组分体系的光谱图，1、2、3 为三个纯组分光谱，其中 3 作为未知干扰光谱；4 为混合物光谱。

(b) 只考虑 1、2 两组分校正模型的 Kalman 滤波的新息序列。

【基本思路】

为了防止模型错误而导致的 Kalman 滤波算法估计失败的情况，人们可采用自适应技术。一般的 Kalman 滤波都是用不变的 $r(k)$ 来进行计算，而自适应 Kalman 滤波方法就是通过在滤波过程中调节 $r(k)$ 来抵制模型错误，它的估计可以由下式给出[104]：

$$r(k)=1/m[\sum v(k-j)v(k-j)]-\boldsymbol{x}(k)'\boldsymbol{P}(k)\boldsymbol{x}(k)$$

式中，m 是新息序列求和的窗口大小。上式通过前 m 次所获得的新息序列来估计 $r(k)$ 的大小。而 $r(k)$ 的大小对滤波算法的影响可由表 6-10 列出的 Kalman 增益估计式看出，当校正模型正确时，$r(k)$ 就很小，这时表 6-10 列出的状态估计式对新的量测信息很敏感，新息序列的变化也相应很小。当模型出现错误时，$r(k)$ 将很快变大，从而使 Kalman 增益变小，使得由表 6-10 列出的状态估计式对新的量测信息不敏感。这样通过估计 $r(k)$ 如此进行的自适应调整，使得 Kalman 滤波算法甚至在模型错误的条件下，亦即在有未知干扰的灰色分析体系的情况下，也可较正确地估计体系待测物种的浓度。

自适应 Kalman 滤波算法除能得到准确的浓度估计以外，通过自适应调整，它还可能在量测过程中得到正确的校正模型。因为 $r(k)$ 实际上是一计算的方差，而这方差正好代表了在第 k 量测点上的模型误差。这样，一个新的校正矢量 $\boldsymbol{x}(k)$ 可以通过 $r(k)$ 计算出来，对于本节讨论的含未知干扰的校正模型，无非是要找到背景干扰的量测系数，即在原模型的校正矢量 $\boldsymbol{x}(k)$ 中增加一个新分量

$$\boldsymbol{x}(k)_{n+1}=b(k)[r(k+m/2)]^{1/2}\qquad 如果\ b(k)>0 \tag{6-131a}$$

$$\boldsymbol{x}(k)_{n+1}=0\qquad 如果\ b(k)<0 \tag{6-131b}$$

式中，n 为原校正模型的组分数，所以 $\boldsymbol{x}(k)_{n+1}$ 实际就是背景干扰的量测系数。式中用 $r(k+m/$

2)实际上就是要利用第 k 量测点附近的 m 个新息序列数据来进行调节，另外，式中的 $b(k)$定义如下：

$$b(k)=1 \qquad 如果\sum v(k-j+m/2)/m>0 \tag{6-132a}$$

$$b(k)=-1 \qquad 如果\sum v(k-j+m/2)/m<0 \tag{6-132b}$$

式（6-128）两式的目的是为了决定校正模型偏差的符号，而这符号则是由 m 个新息系列的数据的平均值来决定的。式(6-127)的采用是因为一般的多组分校正模型量测系数都是正值，即组分浓度的响应一般都是与浓度成正比的。

【基本算法】

从以上对自适应 Kalman 滤波的介绍可以看出，误差协方差阵 $\boldsymbol{P}(k)$的初始值的选择与常规 Kalman 滤波算法是相同的。只要给出 $\boldsymbol{c}(k)$和 $\boldsymbol{P}(k)$的初始值，即 $\boldsymbol{c}(0)$和 $\boldsymbol{P}(0)$，就可按式（6-128）计算其增益矢量，然后按式（6-130）和式（6-129）计算下一个 $\boldsymbol{c}(k)$和 $\boldsymbol{P}(k)$，当到一定步骤(递归次数至少大于 m 时)，用式(6-131a)或式(6-131b)来计算背景干扰$\boldsymbol{x}(k)_{n+1}$。所以，自适应 Kalman 滤波具体算法可由下列步骤组成：

（1）置初值　$\boldsymbol{c}(0)=\boldsymbol{0}$，$\boldsymbol{P}(0)=\gamma^2\boldsymbol{I}$(矢量 $\boldsymbol{c}(0)$中的零元素个数与待测组分个数相同)，这里 $\boldsymbol{I}$ 为单位矩阵，γ^2 为初始估计的量测方差，此值按式(6-133)经验式估出

$$\gamma^2=a\{r(1)/[\boldsymbol{x}(1)^t\boldsymbol{x}(1)]\}^{1/2} \tag{6-133}$$

式中，a 为一与计算机精度相关的因子，一般可取 10～100。值得提出的是，$\boldsymbol{P}$（0）取值太小，可导致有偏估计，如取值太大，则尽管参量的估计值是无偏的，而由于量测点数不足够大时，$\boldsymbol{P}$（0）的最终估计值仍会偏大。

（2）确定 m，一般定在 10～20 之间均可。

（3）启动计算

$$\boldsymbol{g}(k)=\boldsymbol{P}(k-1)\boldsymbol{x}(k)[\boldsymbol{x}(k)^t\boldsymbol{P}(k-1)\boldsymbol{x}(k)+r(k)]^{-1} \tag{6-134a}$$

$$\boldsymbol{c}(k)=\boldsymbol{c}(k-1)+\boldsymbol{g}(k)[z(k)-\boldsymbol{x}(k)^t\boldsymbol{c}(k-1)] \tag{6-134b}$$

$$\boldsymbol{P}(k)=[\boldsymbol{I}-\boldsymbol{g}(k-1)\boldsymbol{x}(k)^t]\boldsymbol{P}(k-1)[\boldsymbol{I}-\boldsymbol{g}(k-1)\boldsymbol{x}(k)^t]^t+\boldsymbol{g}(k-1)r(k)\boldsymbol{g}(k-1)^t \tag{6-135}$$

式中，$r(k)$是量测噪声 $e(k)$的方差，可根据实际量测噪声方差具体确定。重复此步计算直至递归次数大于 m。

（4）利用下述计算式浓度估计未知干扰谱

$$\boldsymbol{x}(k)_{n+1}=b(k)[r(k+m/2)]^{1/2} \qquad 如果\ b(k)>0 \tag{6-136a}$$

$$\boldsymbol{x}(k)_{n+1}=0 \qquad 如果\ b(k)<0 \tag{6-136b}$$

其中

$$b(k)=1 \qquad 如果\sum v(k-j+m/2)/m>0$$

$$b(k)=-1 \qquad 如果\sum v(k-j+m/2)/m<0$$

将未知干扰谱包括于 $\boldsymbol{x}(k)$中，重复上述各步计算，直至达到稳定，即 $\boldsymbol{P}(k)$阵的对角元素在一段计算步骤中皆为很小，此时所得到的 $\boldsymbol{c}(k)$ 即为所求。

【讨论】

在自适应 Kalman 滤波算法中，误差协方差阵 $\boldsymbol{P}$（k）的初始值的选择与常规 Kalman 滤波算法是相同的。在自适应 Kalman 滤波算法中，$\boldsymbol{P}(k)$阵的计算受 $\boldsymbol{r}(k)$ 的影响很大（参见表 6-10 列出的 Kalman 增益估计式和误差协方差阵估计式）。当模型误差很小时，$r(k)$ 是减小的，这将使 Kalman 增益变大，从而导致 $\boldsymbol{P}(k)$ 阵对角元素变小。因为 r（k）值直接与模型的质量相关，所以，自适应滤波的 $\boldsymbol{P}(k)$ 阵的对角元素应该是浓度参数最终估计质量好坏的一个有用

指标。

自适应 Kalman 滤波算法用于灰色分析体系的解析中，存在两个重要的先决条件：① 自适应 Kalman 方法用于分析时，在开始滤波时必须有一段过程校正模型是正确的，亦即在这一过程中，每一个已知存在的待测组分都必须有响应，而背景干扰却无响应；② 对于式（6-131）给出的模型错误校正项只有当背景干扰是一单组分时，$x(k)_{n+1}$就是该组分的量测矢量，但是，因为背景干扰一般来说很难确定其组成，所以，$x(k)_{n+1}$一般只能看成是未包括在校正模型的整个背景干扰的量测系数。在以上两个先决条件中，第一条为一很强的条件，这是因为灰色分析体系的背景干扰实际上是未知的，不能保证在开始滤波时必须有一段过程校正模型是正确的，这就限制了该算法在实际中应用。

（四）局部曲线拟合法

【基本思路】

局部曲线拟合法由 Karstang 和 Kvalheim 提出[105]，其主要思路是利用背景光谱的极大值来进行校正，因为如对光谱进行微分后，背景光谱的极大值处将为零，即在这一波长点上背景干扰将不对待测物的定量测定带来影响，从而达到消除未知背景干扰的效果。问题是因背景干扰是完全未知的，怎样通过已知条件来正确确定背景光谱的极大值的波长位置将是该法成功与否的关键。

对于一般的灰色分析体系的矢量数学模型，均可简化为以下只含一个待测分析物的情况，即

$$\boldsymbol{y}=c_1\boldsymbol{x}_1+\boldsymbol{r}^{\prime}+\boldsymbol{e} \tag{6-137}$$

式中，$\boldsymbol{y}$ 为含未知干扰的混合物光谱量测矢量，$\boldsymbol{x}_1$ 为待测分析物的标准光谱，c_1 为该待测分析物的相对浓度，$\boldsymbol{r}^{\prime}$ 为一假设的未知干扰的量测光谱。必须指出的是，$\boldsymbol{r}^{\prime}$ 这一假设的未知干扰的量测光谱实际上是其他的待测物的量测矢量与实际未知背景干扰量测光谱矢量之和，即

$$\boldsymbol{r}^{\prime}=\sum_{i=2}^{n}c_i\boldsymbol{x}_i+\boldsymbol{r} \tag{6-138}$$

这样，一般的灰色分析体系的数学模型就转化成为了只含一个待测分析物的灰色分析体系模型。这样的变换并没有改变问题的实质，可以通过不断的改变不同的分析物以达到为每一个待测组分定量的目的。下面对只含一个待测分析物的情况进行讨论，其他分析物的定量可完全参照此例同样进行。

注意到 $\boldsymbol{y}$ 是一个量测光谱矢量，所以，它是一个波长的函数，如果对其进行波长方向的微分，可得

$$\mathrm{d}\boldsymbol{y}/\mathrm{d}\lambda=c_1\mathrm{d}\boldsymbol{x}_1/\mathrm{d}\lambda+\mathrm{d}\boldsymbol{r}^{\prime}/\mathrm{d}\lambda \tag{6-139}$$

式中，$\mathrm{d}\boldsymbol{y}/\mathrm{d}\lambda=(\mathrm{d}y_1/\mathrm{d}\lambda,\mathrm{d}y_2/\mathrm{d}\lambda,\cdots,\mathrm{d}y_m/\mathrm{d}\lambda)^{\mathrm{t}}$；$\mathrm{d}\boldsymbol{x}_1/\mathrm{d}\lambda=(\mathrm{d}x_{11}/\mathrm{d}\lambda,\mathrm{d}x_{12}/\mathrm{d}\lambda,\cdots,\mathrm{d}x_{1m}/\mathrm{d}\lambda)^{\mathrm{t}}$；$\mathrm{d}\boldsymbol{r}^{\prime}/\mathrm{d}\lambda=(\mathrm{d}r_1^{\prime}/\mathrm{d}\lambda,\mathrm{d}r_2^{\prime}/\mathrm{d}\lambda,\cdots,\mathrm{d}r_m^{\prime}/\mathrm{d}\lambda)^{\mathrm{t}}$；$m$ 是量测光谱的波长点数。

设在假设的未知干扰的量测光谱存在有一个极大点，记为第 k 点，则有

$$\mathrm{d}y_k/\mathrm{d}\lambda=c_1\mathrm{d}x_{1k}/\mathrm{d}\lambda+\mathrm{d}r_k^{\prime}/\mathrm{d}\lambda \tag{6-140}$$

因 $\mathrm{d}r_k^{\prime}/\mathrm{d}\lambda$ 在极大点为零，所以有

$$c_1=(\mathrm{d}y_k/\mathrm{d}\lambda)/(\mathrm{d}x_{1k}/\mathrm{d}\lambda) \tag{6-141}$$

也就是说，只要找到了未知干扰的量测光谱的一个极大点的波长位置，就可利用这一点来求得待测物的相对浓度。问题是怎样来正确估计出未知干扰的量测光谱极大点的波长位置。该法的几何意义示于图 6-14。

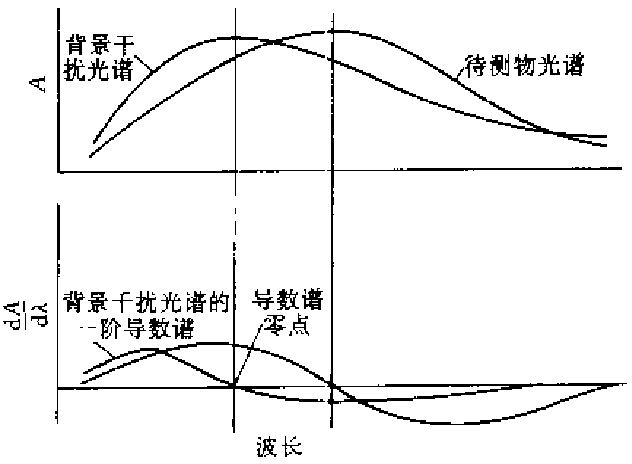


图 6-14 利用一阶导数求待测物浓度示意图

在文献[105]中，Karstang 介绍了一种基于同时利用一阶导数和三阶导数来估计未知干扰的量测光谱极大点的波长位置方法，在此将介绍一种基于迭代目标因子分析法来估计未知干扰的量测光谱极大点波长位置的方法，找到这个极大点后，继用式(6-141)即可求得待测物的相对浓度。

【基本算法】

从上面介绍的基本思路可知，局部曲线拟合法具体可由以下步骤来实现：

(1) 用 $\boldsymbol{x}_1$ 构成一个正交投影矩阵 $\boldsymbol{P}$，

$$\boldsymbol{P}=(\boldsymbol{I}-\boldsymbol{x}_1\boldsymbol{x}_1^t) \qquad (6\text{-}142)$$

式中，$\boldsymbol{I}$ 为一 $(m\times m)$ 阶的单位矩阵。

(2) 用正交投影矩阵 $\boldsymbol{P}$ 对混合物量测光谱进行投影，得到一与 $\boldsymbol{x}_1$ 正交的，即属于 $\boldsymbol{x}_1$ 矢量零空间的一个矢量 $\boldsymbol{y}^*$，该矢量已将 $\boldsymbol{x}_1$ 的影响全部扣除，主要反应了未知干扰的量测光谱的信息；

$$\begin{aligned}\boldsymbol{Py}&=[\boldsymbol{I}-\boldsymbol{x}_1(\boldsymbol{x}_1^t\boldsymbol{x}_1)^{-1}\boldsymbol{x}_1^t]\boldsymbol{y}\\&=[\boldsymbol{I}-\boldsymbol{x}_1(\boldsymbol{x}_1^t\boldsymbol{x}_1)^{-1}\boldsymbol{x}_1^t](c_1\boldsymbol{x}_1+\boldsymbol{r}^s)\\&=[\boldsymbol{I}-\boldsymbol{x}_1(\boldsymbol{x}_1^t\boldsymbol{x}_1)^{-1}\boldsymbol{x}_1^t]\boldsymbol{r}^s=\boldsymbol{y}^*\end{aligned}$$

(3) 用 $\boldsymbol{x}_1$ 和 $\boldsymbol{y}$ 构成一个矩阵 $\boldsymbol{Y}$，即 $\boldsymbol{Y}=[\boldsymbol{y},\ \boldsymbol{x}_1]$，并对其进行主成分分解，

$$\boldsymbol{Y}=\boldsymbol{USV}^t=\boldsymbol{UT}^t \qquad (6\text{-}143)$$

在此，$\boldsymbol{T}=\boldsymbol{SV}^t$，常称为因子载荷矩阵，$\boldsymbol{U}$ 则称为因子得分矩阵，继用它构成迭代用投影矩阵，对 $\boldsymbol{y}^*$ 进行迭代运算：

令 $\boldsymbol{r}^{(0)}=\boldsymbol{y}^*$，$k=1,2,\cdots$

$$\boldsymbol{r}^{(k)}=\boldsymbol{UU}^t\boldsymbol{r}^{(k-1)} \qquad (6\text{-}144)$$

在每次迭代时，先将 $\boldsymbol{r}^{(k-1)}$ 中的负元素都置为零，然后再进行投影运算，这样的迭代过程一直迭代到 $\|\boldsymbol{r}^{(k)}-\boldsymbol{r}^{(k-1)}\|$ 小于某一给定值。这样迭代求得的背景矢量将会很靠近真实的背景，此时，再找到该背景的极大值的波长位置，用式（6-141）来求得待测分析物的相对浓度。

【讨论】

一般说来，该法求出的待测分析物的相对浓度，在背景光谱与分析物光谱重叠度不大时结果将与实际值比较靠近，因此时背景光谱极大值的波长位置估计将比较准确，然而，如果背景光谱与分析物光谱重叠度很大时结果将不太可靠了。从这一角度来看，虽用式(6-141)可得到唯一解，但背景光谱极大值的波长位置估计是否准确仍受到其他因素的制约。为了得到较精确解，一般需要采用内插法来求真正极大点的位置，实因极大点的位置可能实际是落在两个量测波长点之间，这也是为什么将其称为局部曲线拟合方法的原因。综上所述，实际上此法仍只能说是得到了一个可能解，真正得到严格唯一解的还需用下一节讨论的二维数据的化学计量学方法才能得到。

二、矩阵校正方法

从上几节的讨论中可以看到，用矢量校正模型来进行灰色分析体系的解析，都存在很强

的先决条件，如果这些先决条件不能满足，则很难保证求得的解是有物理意义的真实解。可以认为，这样的困难主要是来自对未知干扰背景谱知之甚少，它可以是由任意的很多未知干扰所组成，而对于这样的背景干扰我们又根本无法从一个矢量中确定其组分数。先来看一个最简单的灰色分析体系的例子，即在这样的灰色分析体系中，定性已知的待测组分只有一个，其数学模型为

$$y=c_1x_1+r+e \tag{6-145}$$

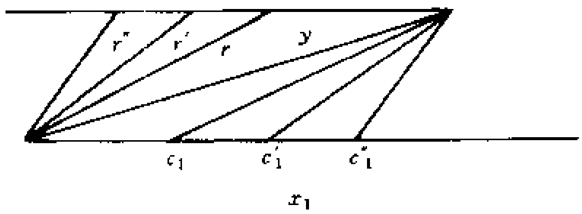


图 6-15　不同待测物的浓度可得不同未知背景干扰谱的矢量示意图

c_1、c_1'、c_1''分别表示不同的待测物浓度；r、r'、r''则表示由此而得的不同的未知背景干扰光谱矢量，它们都可以与 x_1 很好地拟合混合物光谱 y

在此模型中，代求的参数只有 c_1。当然，背景干扰量测谱 $\boldsymbol{r}$ 还是未知的。这可以说是灰色分析体系中一种最简单的情况。在这里，除可以测得该混合物试样的量测矢量 $\boldsymbol{y}$（包括矢量的方向和长度）以外，还可得到待测物的标准光谱 $\boldsymbol{x}_1$ 的形状（即标准谱矢量的方向），实因 $\boldsymbol{x}_1$ 在量测矢量 $\boldsymbol{y}$ 中的贡献（即标准谱矢量的长度）应由 c_1 来确定，这种情况的几何意义可由图 6-15 来表示。

从图 6-15 可以看出，由于背景干扰量测谱 $\boldsymbol{r}$ 和 c_1 皆为未知，c_1 和 $\boldsymbol{r}$ 又互为函数，对应不同的 c_1 就可得到不同的背景干扰量测谱 $\boldsymbol{r}$，只要可保持 c_1 和 $\boldsymbol{r}$ 皆为非负就都是可能解。值得指出的是，在这种情况下，任何拟合技术都是没有意义的，拟合度的好坏不能成为评价结果的标准。图 6-16 给出了一个这种情况的具体计算机模拟例了。从图 6-16 可以看出，随着 c_1 的变化，背景干扰谱也在不断的变化。用这样不同的 c_1 和 $\boldsymbol{r}$ 都可以把混合试样的量测矢量 $\boldsymbol{y}$ 拟合好。

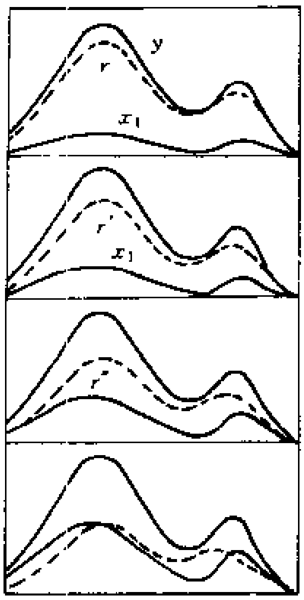


图 6-16　由图 6-15 所示最简单灰色体系的计算机模拟示例

其中背景干扰光谱由点线表示出

如果采用矩阵类型的数据，情况将发生显著的变化。对于灰色分析体系，矩阵校正的数学模型可由式（6-146）表示：

$$\boldsymbol{Y}=c_1\boldsymbol{X}_1+c_2\boldsymbol{X}_2+\cdots+c_n\boldsymbol{X}_n+\boldsymbol{R}+\boldsymbol{E} \tag{6-146}$$

式中，$\boldsymbol{Y}$ 表示混合物的量测矩阵（可为二维荧光光谱或色谱联用仪产生的二维数据）；$\boldsymbol{X}_i$（$i=1, 2, \cdots, n$）为待测组分纯物种的量测矩阵。值得提出的是，像 $\boldsymbol{X}_i$（$i=1, 2, \cdots, n$）这样的矩阵，一般为双线性矩阵，即它们可由两个矢量的外积表示

$$\boldsymbol{X}_i=\boldsymbol{s}_i\boldsymbol{c}_i^t \tag{6-147}$$

从式（6-147）可知，$\boldsymbol{X}_i$ 的秩为 1，二维荧光光谱或色谱联用仪产生的二维数据均可满足这一条件。矩阵校正的数学模型中的 $\boldsymbol{E}$ 为量测误差矩阵，一般仍假设为服从正态分布的等方差白噪声误差；n 为混合体系中的待测物种数，c_i（$i=1, 2, \cdots, n$）为未知待估参数，一般可看成为相对浓度；$\boldsymbol{R}$ 为未知背景干扰矩阵，它亦可以是多个未知干扰物的量测矩阵的线性加合，即

$$\boldsymbol{R}=d_1\boldsymbol{B}_1+d_2\boldsymbol{B}_2+\cdots+d_m\boldsymbol{B}_m \tag{6-148}$$

式中，$\boldsymbol{B}_i$（$i=1, 2, \cdots, m$）为干扰纯物质的量测矩阵；m 为干扰物种数；d_i（$i=1, 2, \cdots,$

m）为干扰物的相对浓度。

双线性矩阵的特点，即纯物质的量测矩阵的秩为1，而背景干扰量测矩阵的秩就代表它的组分数，也就是这一特点，使得灰色分析体系的矩阵校正方法成为近年来分析化学计量学研究的热点和难点，并已取得较大的成果，其中著名的有广义秩消失因子分析法（GRAFA），残差双线性分解法（RBL），和约束背景双线性分解法（CBBL）。从纯数学的角度看，因为二维数据提供了矩阵秩的新信息，这些方法一般皆可给出有物理意义的唯一解。当然，在实际应用中，还存在一些具体困难。本节将对这些方法的基本思路和基本算法作出较详细的介绍。

（一）秩消失因子分析法

【基本思路】

秩消失因子分析法首先由 C.-N. Ho，Christian 和 Davidson 于1978年针对二维荧光谱的定量分析而提出的[106~108]。该方法的基本原理就是利用纯物种二维荧光光谱是一个双线性矩阵，且其秩为一的特点来进行的。此方法的基本思路为，设已测得被分析物的纯物种二维谱，将其作为标准，进而对含未知干扰的混合体系的二维荧光光谱来进行消去该标准的运算。设未进行消去运算前，含未知干扰的混合体系的量测矩阵的秩为 k，那么，当被消去的标准正好等于其存在于混合体系的该物种的浓度大小时，此时未知干扰的混合体系的量测矩阵的秩就会减少一而成为 $(k-1)$，据此即可对该被分析物进行定量。

设含未知干扰的混合体系的量测矩阵为 $\boldsymbol{Y}$，不失一般性，设本章所论及的灰色分析体系，可简化为如下的数学模型：

$$\begin{aligned}\boldsymbol{Y} &= c_1\boldsymbol{X}_1+\boldsymbol{R}+\boldsymbol{E}\\ &= c_1\boldsymbol{p}_1\boldsymbol{q}_1^{t}+\boldsymbol{R}+\boldsymbol{E}\end{aligned} \tag{6-149}$$

即先只考虑其中一个感兴趣的分析物，而把其他分析物也都归于未知干扰。这样的简化不影响一般性的原因，是因为我们可连续地将不同的分析物当成待分析物，直至所有的分析物都被测定。式中，$\boldsymbol{X}_1$ 即为该纯物种双线性的量测矩阵；$\boldsymbol{p}_1$ 为其纯物质荧光激发谱（对于色谱联用仪器则为纯物种色谱）；而 $\boldsymbol{q}_1^{t}$ 为其纯物质荧光发射谱（对于色谱联用仪器则为纯物种的相应光谱）；c_1 是其相对浓度；$\boldsymbol{R}$ 为未知背景干扰，与式（6-146）不同的地方是在此处它是由其他分析物即多个未知干扰物的量测矩阵的线性加合而成；$\boldsymbol{E}$ 为二维数据阵的量测误差矩阵，一般仍设为白噪声。

首先，考察一下秩消失因子分析法的一个最主要方程：

$$\boldsymbol{F}=\boldsymbol{Y}-\beta\boldsymbol{X}_s \tag{6-150}$$

式中，$\boldsymbol{X}_s$ 为定性已知的与 $\boldsymbol{X}_1$ 相对应的待测分析物的标准物量测矩阵，它可以表示为

$$\boldsymbol{X}_s=c_s\boldsymbol{p}_s\boldsymbol{q}_s^{t} \tag{6-151}$$

为以后讨论方便，还假设 $\boldsymbol{p}_s$ 和 $\boldsymbol{q}_s^{t}$ 分别是标准化的矢量，即满足 $\sum p_{sk}^2=1(k=1,\ 2,\ \cdots,\ w)$，（在此 w 为激发谱的量测该长点数或色谱保留时间点数）和 $\sum q_{sl}^2=1(l=1,\ 2,\ \cdots,\ r)$，（在此 w 为发射谱的量测波长点数或与色谱联用的相应光谱的量测波长点数）；秩消失因子分析法的主要目的就在于找到 β，即找到 $\beta=c_1/c_s$，从而使此时矩阵 $\boldsymbol{F}$ 的秩比原矩阵 $\boldsymbol{Y}$ 的秩少一。这也就是秩消失因子分析法的主要思路。因当时计算矩阵秩的变化还不十分容易，故 C.-N. Ho，Christian和 Davidson 是采用一种迭代方法来实现这一计算的。

【基本算法】

（1）对 $\boldsymbol{Y}$ 矩阵进行主成分分析（即奇异值分解）

$$\boldsymbol{Y}=\boldsymbol{USV}^{t} \tag{6-152}$$

式中，U 矩阵收集了 Y 矩阵的所有非噪声左特征矢量，V^t 矩阵收集了 Y 矩阵的所有非噪声的右特征矢量，一般说来，Y 矩阵中所含的化学物种数为多少就得收集多少个特征矢量。如设 Y 矩阵中所含的化学物种数为 n，则 U 和 V 分别为（$w\times n$）和（$n\times r$）阶矩阵。

Y 矩阵的阶数是由激发谱的量测波长点数（w）和发射谱的量测波长点数（r）来决定的，所以，其维数（w 或 r）一般都要大大地大于 Y 矩阵中所含的化学物种数 n。

（2）由式（6-150）定义的 F 矩阵是 β 的函数，所以，秩消失因子分析法的秩运算是针对 F 矩阵来进行的。然而，因 F 矩阵一般都比较大，故由 C.-N. Ho，Christian 和 Davidson 提出的秩消失法实际上是针对下述两矩阵来进行的。设

$$G_1=V^tF^tFV$$

或

$$G_2=U^tFF^tU$$

这样，就可将矩阵 F 的维数从 w（或 r）降到 Y 矩阵中所含的化学物种数（n）一样大，从而大大地减少了计算量。进行这样简化是有理由的，这是因为 U 与 Y 的列矢量张成同一线性空间，而 V^t 与 Y 的行矢量张成同一线性空间，这样的主成分投影降低了数据的维数，但却不丢失任何有用的信息。

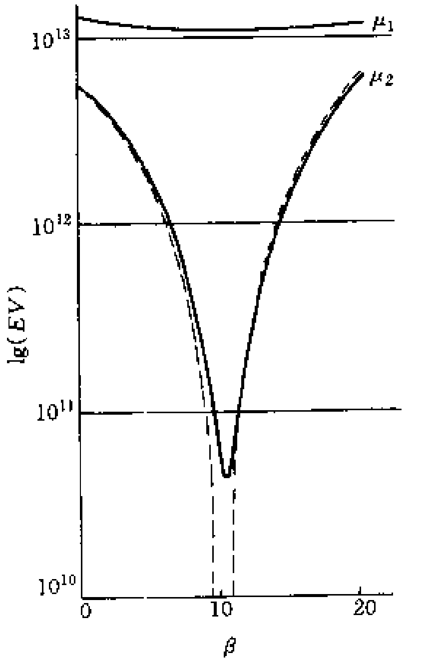


图 6-17　作为 β 的函数之第一特征值（μ_1）和第二特征值（μ_2）变化图

注：该体系为两组分分析体系，其中一个为分析物，另一个为未知干扰物。

（3）变化式（6-150）中的 β 之值，变化一般是从零开始向正方向变化。对不同的 β 计算 **G1**（或 **G2**）矩阵的特征值，一般说来当其中最小的那个特征值达到极小（亦即靠近零）时，此时所对应的 β 值就是我们所要求的值。

（4）因为 $\beta=c_1/c_s$，而 c_s 是标准物的量测矩阵的相对浓度，是已知的，所以，c_1 就等于 βc_s。

因这一方法巧妙地利用了二维数据的秩的信息，为解决分析化学中灰色体系的直接快速分析提供了一条新的途径，受到了化学计量学家们的普遍关注，并将其拓广至联用色谱二维数据的解析[109,110]。利用该法来求得 β 的过程可从图 6-17 清楚看出，随着 β 的不断增大，第二个特征值正好在 c_1/c_s 点处出现一个极小值说明在此点上，矩阵的秩为 1，然而，随着 β 的继续增大，第二个特征值又回升，说明此时矩阵的秩又回复到 2。

【讨论】

C.-N. Ho，Christian 和 Davidson 提出秩消失法的同时，还提出了 Bessel 不等式的概念，以检验在含未知干扰的混合体系的量测矩阵 Y 中是否确实含有待测标准物，似有必要在本节也作出介绍。

因为 U 与 Y 的列矢量张成同一线性空间，所以，p_s 应可由 U 矩阵的列矢量线性表示，亦即

$$p_s=\sum a_k u_k \quad (k=1,\cdots,n) \tag{6-153}$$

同理，因为 V^t 与 Y 的行矢量张成同一线性空间，所以，q_s 应可由 V 矩阵的列矢量线性表示，亦即

$$q_s=\sum b_k v_k \quad (k=1,\cdots,n) \tag{6-154}$$

因，U 与 V 都是正交归一化矩阵，即，$U'U=I$ 和 $V'V=I$ 成立，所以

$$a_k=u_k^{t}p_s \quad (k=1,\cdots,n) \tag{6-155a}$$

及

$$b_k=v_k^{t}q_s \quad (k=1,\cdots,n) \tag{6-155b}$$

如果把 a_k（$k=1$，…，n）和 b_k（$k=1$，…，n）写成矢量的形式，则有

$$a=U^{t}p_s \tag{6-156a}$$

及

$$b=V^{t}q_s \tag{6-156b}$$

于是有

$$\begin{aligned}\sum a_k^2 &= a^{t}a=p_s^{t}UU^{t}p_s \quad (k=1,\cdots,n)\\ &=p_s^{t}p_s\\ &=\sum p_{si}^2=1 \quad (i=1,\cdots,w)\end{aligned} \tag{6-157a}$$

在式（6-157）推导的第三步中，利用了投影矩阵 UU^{t} 的特性，即 $UU^{t}p_s=p_s$，这是因为U^{t} 就是 U 的广义逆，即 $UU^{t}U=U$，这样，UU^{t} 就构成了一个包含 p_s 矢量的投影矩阵（参见第十一章）。同时因$\sum p_{si}^2=1$ 的条件，所以有$\sum a_k^2=1$ 的结论成立。同理，因为

$$\begin{aligned}\sum b_k^2 &= b^{t}b=q_s^{t}VV^{t}q_s \quad (k=1,\cdots,n)\\ &=q_s^{t}q_s\\ &=\sum q_{si}^2=1 \quad (i=1,\cdots,r)\end{aligned} \tag{6-157b}$$

再利用$\sum q_{si}^2=1$（$i=1$，…，r）的条件，也有$\sum b_k^2=1$ 结论成立。

也就是利用了式$\sum a_k^2=1$ 和式$\sum b_k^2=1$，得到了所谓的 Bessel 不等式。

$$(\sum a_k^2)\leqslant 1 \tag{6-158a}$$

或

$$(\sum b_k^2)\leqslant 1 \tag{6-158b}$$

在此，将$\sum b_k^2=\sum q_{si}^2=1$ 改成式（6-158）两不等式，其理由在于 U 和 V 只是取了主成分，没有将量测误差考虑在内。所以，在化学量测的实际情况中，成立的只是不等式，因为在实际情况中，$UU^{t}p_s$ 只能近似的等于 p_s。

根据以上讨论，可以得出这样的结论，即（$\sum a_k^2$）或（$\sum b_k^2$）越是靠近 1，就说明 X_s 的确存在于含未知干扰的样本量测矩阵 Y 之中。利用 Bessel 不等式，可检验在含未知干扰的混合体系的量测矩阵 Y 中是否确实含有待测标准物。

因以上讨论的秩消失因子分析法实质上是一种需要多次或迭代的算法，故 Lorber 对此方法作出了改进，他在引入奇异值分解算法的基础上，提出一种非迭代式的秩消失因子分析法。Lorber 的非迭代式的秩消失因子分析法[111]的主要思路就是将以上讨论的秩消失因子分析法转化成为一个特征方程的问题，即将式（6-150）转化成为以下的特征方程

$$X^{*}z=\lambda z \tag{6-159}$$

的形式，从而使秩消失因子分析法的计算变得简单明了。因 Lorber 的非迭代方法实际上就是广义秩消失因子分析法的第一种情况，此方法在此暂不作介绍，留到下一节一并讨论。

（二）广义秩消失因子分析法

广义秩消失因子分析法是由 Sanchez 和 Kowalsky 于 1986 年提出[112]，其主要思路是将秩消失因子的计算问题转变成为广义特征方程求解的问题，从而使得秩消失因子分析法不再局限于一次只能测一个待测分析物的约束，此外，该法还可拓广至其他类型的情况。以下将对

该法作出详细介绍。

【广义秩消失因子分析法的基本公式和不同的四种类型】

任一双线性数据矩阵 $\boldsymbol{Y}$ 都能表为它所含的 n 个纯物种双线性数据矩阵 $\boldsymbol{X}_i$ 的线性组合，即

$$\begin{aligned}\boldsymbol{Y} &= \lambda_1\boldsymbol{X}_1 + \lambda_2\boldsymbol{X}_2 + \cdots + \lambda_n\boldsymbol{X}_n - \boldsymbol{E} \\ &= \lambda_1\boldsymbol{p}_1\boldsymbol{q}_1^t + \lambda_2\boldsymbol{p}_2\boldsymbol{q}_2^t + \cdots + \lambda_n\boldsymbol{p}_n\boldsymbol{q}_n^t - \boldsymbol{E} \\ &= \boldsymbol{P\lambda Q}^t \end{aligned} \tag{6-160}$$

在此

$$\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_n] \tag{6-161}$$

$$\lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \lambda_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \lambda_n \end{bmatrix} \tag{6-162}$$

$$\boldsymbol{Q}^t = \begin{bmatrix} \boldsymbol{q}_1^t \\ \boldsymbol{q}_2^t \\ \vdots \\ \boldsymbol{q}_{n-1}^t \\ \boldsymbol{q}_n^t \end{bmatrix} \tag{6-163}$$

一般说来，对于秩消失因子分析法存在两个矩阵，一个是含未知干扰的 $\boldsymbol{Y}$ 数据矩阵，另一个则是含定性已知的标准分析物的 $\boldsymbol{X}$ 矩阵，它也可表为

$$\begin{aligned}\boldsymbol{X} &= \xi_1\boldsymbol{X}_1 + \xi_2\boldsymbol{X}_2 + \cdots + \xi_n\boldsymbol{X}_n + \boldsymbol{E} \\ &= \xi_1\boldsymbol{p}_1\boldsymbol{q}_1^t + \xi_2\boldsymbol{p}_2\boldsymbol{q}_2^t + \cdots + \xi_n\boldsymbol{p}_n\boldsymbol{q}_n^t + \boldsymbol{E} \\ &= \boldsymbol{P\xi Q}^t \end{aligned} \tag{6-164}$$

式中

$$\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_n] \tag{6-165}$$

$$\xi = \begin{bmatrix} \xi_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \xi_2 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \xi_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \xi_n \end{bmatrix} \tag{6-166}$$

$$\boldsymbol{Q}^t = \begin{bmatrix} \boldsymbol{q}_1^t \\ \boldsymbol{q}_2^t \\ \vdots \\ \boldsymbol{q}_{n-1}^t \\ \boldsymbol{q}_n^t \end{bmatrix}$$

由(6-160)和(6-164)两式表示的矩阵 $\boldsymbol{Y}$ 和 $\boldsymbol{X}$ 似乎完全一样，可实际上却是不同的，它们不同的地方在于 $\boldsymbol{\lambda}$ 和 $\boldsymbol{\xi}$。Sanchez 和 Kowalski 将它们分成四种不同情况来分别处理，为了保持该方法的原始性，分别对这 4 种情况进行讨论，必要时对它们加以相应的说明。

(1) 在定性已知的标准分析物的 $\boldsymbol{X}$ 矩阵中只有一种物质，而且，该物种是含未知干扰的样

本数据矩阵 Y 中的一个，即

$$\mathrm{Diag}(\xi)=(\xi_1,0,\cdots,0)$$

$$\mathrm{Diag}(\lambda)=(\lambda_1,\lambda_1,\cdots,\lambda_n)\quad(n>1)$$

这种情况也就是在前一节中所讨论的那种情况。

(2) 在定性已知的标准分析物的 X 矩阵中含有几种物质，而且，这些物种都存在于样本数据矩阵 Y 中，即

$$\mathrm{Diag}(\xi)=(\xi_1,\cdots,\xi_m,0,\cdots,0)$$

$$\mathrm{Diag}(\lambda)=(\lambda_1,\lambda_2,\cdots,\lambda_n)\quad(n>m)$$

这种情况也就是在本书中定义的灰色分析体系的一般情况。

(3) 在定性已知的标准分析物的 X 矩阵中含有 n 种物质，但是，这些物种不一定都存在于样本数据矩阵 Y 中，即

$$\mathrm{Diag}(\xi)=(\xi_1,\cdots,\xi_m,\cdots,\xi_n)$$

$$\mathrm{Diag}(\lambda)=(\lambda_1,\cdots,\lambda_m,0,\cdots,0)\quad(m<n)$$

这种情况实质上相当于在本书中定义的白色分析体系的一种特殊情况，即校正矩阵中已包含了所有待测物的信息，与第三章中所讨论的情况约为不同之点在于此时还多包含了其他化学物种的标准谱信息，且这些物种在样本量测矩阵中不存在，此时原则上是可采用矢量数据的校正方法来直接求解的，文献[113~115]提供了几种可能解法，有兴趣的读者可参阅这些文献。值得提出的是，对于这样的情况，如果简单采用多元线性回归的方法来求解的话，会出现负浓度值的问题，而使最小二乘估计结果不可靠。一般说来，此时如果采用带约束回归分析方法，即约束所估参数值（在多元校正中即为化学物种的浓度值）不为负数，这样即可直接求出（λ_1，…，λ_m）来，而毋需采用本节将要介绍的需采用二维数据的广义特征问题的求解方法，亦即采用广义秩消失因子分析法来解决此类问题。

(4) 在 X 矩阵中含有 k 种物质，但是，这些物种不一定都存在于数据矩阵 Y 中，而且，在数据矩阵 Y 中所含的物种也不一定都存在于数据矩阵 X 中，即

$$\mathrm{Diag}(\xi)=(\xi_1,\cdots,\xi_m,\cdots,\xi_k,0,\cdots,0)\quad(k<n)$$

$$\mathrm{Diag}(\lambda)=(0,\cdots,0,\lambda_r,\cdots,\lambda_n)\quad(r>1)$$

这种情况，因 X 矩阵本身还包含了一些含未知干扰样本中不存在的物种，所以，很难把它定义为前三种情况中的定性已知的标准矩阵，同时，Y 矩阵亦包含了未知干扰，所以，它实质上应相当于在本书中定义的黑色分析体系的一种特殊情况。值得指出的是，Sanchez 和 Kowalski 提出的广义秩消失因子分析法确可对此求解，说明广义秩消失因子分析法实质上是一种黑色分析体系的解析方法，关于这一点，将在讨论广义秩消失因子分析法的具体解法中给出必要说明。关于广义秩消失因子分析法解析黑色分析体系的具体算法，将在第四节详细介绍。

根据式 $Y\approx P\lambda Q^t$ 和式 $X\approx P\xi Q^t$，对此两方程同时右乘 Q^t 的广义逆 $(Q^t)^+$，可得

$$P\lambda=Y(Q^t)^+\quad 和\quad P\xi=X(Q^t)^+$$

$P\lambda\xi=P\xi\lambda$，是因为 λ 和 ξ 都是对角矩阵，其矩阵乘法是可以交换的，所以可得

$$Y(Q^t)^+\xi=X(Q^t)^+\lambda\tag{6-167}$$

令 $(Q^t)^+=Z$，则广义秩消失因子分析法的主要方程可变为

$$XZ\lambda=YZ\xi\tag{6-168}$$

对于上述方程，已知 X 和 ξ，而且也测得了 Y，所以要求解此方程，必须先求得 Z 和 λ。

由方程（6-168）定义的问题实质上就相当于线性代数中的广义特征值问题（参见第十一章），对于广义特征值，目前只能用 QZ 算法求解 XZ 和 YZ 都是方阵的情况[116]，所以，对于它们的求解将针对不同的情况来具体进行。

【四种不同类型的算法】

1. 广义秩消失因子分析法对第一种情况的求解方法

首先，按式（6-152）对样本矩阵 Y 进行主成分分析（奇异值分解）。

这样，方程（6-168）可变为

$$XZ\lambda=USV^tZ\xi \tag{6-169}$$

令 $Z=VS^{-1}Z^*$，这样，$Z^*=SV^tZ$，所以，式（6-169）可变为

$$XVS^{-1}Z^*\lambda=USV^tZ\xi=USV^tVS^{-1}Z^*\xi=UZ^*\xi \tag{6-170}$$

上述推导中使用了 V 矩阵的正交性质，即 $V^tV=I$，再对式（6-170）左乘 U^t 得

$$U^tXVS^{-1}Z^*\lambda=U^tUZ^*\xi=Z^*\xi \tag{6-171}$$

继续对式（6-171）右乘 λ^{-1}，可得

$$U^tXVS^{-1}Z^*\lambda\lambda^{-1}=U^tXVS^{-1}Z^*=Z^*\xi\lambda^{-1}=Z^*\gamma \tag{6-172}$$

亦即

$$X^*Z^*=Z^*\gamma \tag{6-173}$$

式中，$X^*=U^tXVS^{-1}$，从式（6-172）和式（6-173）可以看出，由式（6-168）表出的广义特征值问题被转换为一般特征值问题，而且，X^* 矩阵还是一个方阵。注意到对于第一种情况，X 是一个秩为一的矩阵，所以 X^* 也是一个秩等于 1 的矩阵。对于一个秩为 1 的矩阵，其唯一的不为零的特征值就是这个矩阵的迹，如果记 U^tp_1 为 a，q_1^tV 为 b，则矩阵 X^* 的迹可表为

$$\gamma=\mathrm{Trace}(X^*)=\mathrm{Trace}(U^tXVS^{-1}) \tag{6-174}$$

同时，$\xi\lambda^{-1*}=\gamma$，亦即 $\gamma=\xi_1/\lambda_1$，在此 γ 就是那个唯一的不为零的特征值，实因其余的 γ 均因 $\xi_i(i=2,\cdots,n)$ 为零而全为零。所以，求得 γ 后，就立即可以求出，即

$$\lambda_1=\xi_1/\gamma \tag{6-175}$$

这也就是要求的含未知干扰样本中的待测物的相对浓度。

上述方法也就是 Lorber 所提出的非迭代式秩消失因子分析法[117]。

2. 广义秩消失因子分析法对第二种情况的求解方法

这种情况与第一种情况的不同之点就在于校正矩阵包含的物种数，第一种情况只含一种物质，而第二种情况是含多种物质。

同样利用式(6-168)，并对方程的两边右乘 λ^{-1}，可得

$$XZ\lambda\lambda^{-1}=XZ=YZ\xi\lambda^{-1}=YZ\beta \tag{6-176}$$

即

$$XZ=YZ\beta \tag{6-177}$$

将式(6-152)代入式(6-177)，得

$$XZ=USV^tZ\beta \tag{6-178}$$

令

$$Z=VS^{-1}Z^*$$

这样，$Z^*=SV^tZ$，式(6-178)可变为

$$XVS^{-1}Z^*=USV^tZ\beta=USV^tVS^{-1}Z^*\beta=UZ^*\beta \tag{6-179}$$

继对式(6-179)两边左乘 U^t，得

$$U^tXVS^{-1}Z^*=Z^*\beta \tag{6-180}$$

由式(6-180)表出的是一个典型的特征值问题，与第一种情况不同的是，在此存在有 m 个而不

只是一个非零特征值,即($\beta_1,\cdots,\beta_m$),为了确定哪一个特征值对应哪一个化学物质,在此需要用到相应的特征向量。利用所得的特征向量,可求出相应的 p_i 和 q_i^t,即相应的激发与发射光谱来,因为

$$Z^* = SV^tZ = SV^t(Q^t)^+$$

$$Q^t = (VS^{-1}Z^*)^+$$

求得 Q^t 后,不难求出相应的 $P\lambda$ 来,因 $Y = P\lambda Q^t = USV^t$,所以

$$P\lambda = Y(Q^t)^+ = USV^tVS^{-1}Z^* = UZ^* \tag{6-181}$$

从上述讨论中可以看出,广义秩消失因子分析法可直接相应的激发与发射光谱,即具有分辨的功效,关于这一点将在第四节详细讨论。另外,因为 $\xi\lambda^{-1}=\beta$,所以求得了 $\beta=(\beta_1,\cdots,\beta_m)$,也就相当于求得了一系列的相对浓度的比值($\xi_1/\lambda_1$,…,$\xi_m/\lambda_m$),因为($\xi_1$,…,$\xi_m$)均为已知,所以,($\lambda_1$,…,$\lambda_m$)就很容易求出了。

3. 广义秩消失因子分析法对第三种情况的求解方法

这种情况与第二种情况的不同之点就在于校正矩阵包含的物种数大于样本数据矩阵包含的物种数,此时如果将 X 和 Y 的位置对换一下,即此时是对 X 而不是对 Y 进行奇异值分解,继用第二种情况的解法来解第三种情况,即可求出 Y 矩阵各组分的相对浓度来。因以下所有求解的步骤都与第二种情况一样,在此不再赘述。

4. 广义秩消失因子分析法对第四种情况的求解方法

这种情况是一种特殊情况。校正矩阵包含的物种没全被样本矩阵所含物种所包含,反之,样本矩阵包含的物种也没全被校正矩阵所含物种所包含,与第二种情况和第三种情况都不同。为使一个矩阵所含物种能包含另一个矩阵的全部物种,所采用的方法是,将校正矩阵 X 和样本矩阵 Y 相加,以得到一个新矩阵 M。这样,这一新得到的矩阵 M 所含物种既包含了 X 矩阵的全部物种又包含了 Y 矩阵的全部物种,用第二种情况中所介绍的方法即可求解了。具体解法如下:

①
$$M = X + Y \tag{6-182}$$

② 对矩阵 M 进行主成分分解

$$M = U_MS_MV_M^t \tag{6-183}$$

③ 构造以 M 矩阵为基底的特征值方程,即

$$(U_M^tYV_MS_M^{-1})Z_M^* = Z_M^*\beta \tag{6-184}$$

④ 求得 Z_M^* 后,再分别用式求得 P 和 Q^t

$$P(\xi+\lambda) = U_MZ_M^* \tag{6-185a}$$

$$Q^t = (V_MS_M^{-1}Z_M^*)^+ \tag{6-185b}$$

【讨论】

对于广义秩消失因子分析法用于灰色分析体系的解析,第一种和第二种情况最重要,其他两种情况只是第二种情况的一种拓展。另外,无论是哪种情况,在具体求解之前,似都有必要进行一些必要的检验,即检验要求的分析物是否确实存在于样本矩阵之中,在此介绍一种基于投影矩阵的简便方法。当然,用前一节所讨论的 Bessel 不等式也可同样进行类似检验。

在讨论灰色分析体系的基于矢量校正的算法中,我们曾多次采用过投影矩阵(参见第十一章)来进行运算,这是因为投影矩阵具有式(6-186)所示的性质。

$$XX^+x_i = x_i \tag{6-186}$$

式中，$X=[x_1,\cdots,x_i,\cdots,x_n]$。

式（6-186）的物理意义为：如果 x_i 包含于 X 之中，则由 X 构成的投影矩阵 XX^+ 作用于 x_i 之上，x_i 将不变，这样就可以采用投影的办法来检验某一物种 x_i 是否确实包含于某一混合物的量测矩阵 X 之中。投影也可以通过式（6-187）进行，

$$UU^tx_i=x_i \tag{6-187}$$

式中，U 矩阵是 X 矩阵的主成分分解的左特征向量矩阵，即

$$X=USV^t \tag{6-188}$$

式（6-187）之所以成立，是因为

$$XX^+=USV^t(USV^t)^+=USV^tVS^{-1}U^t=UU^t \tag{6-189}$$

在此利用了 U 矩阵的正交性质，即 $UU^tU=U$，也就是说，U^t 就是 U 矩阵的广义逆。上式说明，任一与 X 矩阵张成同一线性空间或包含了 X 矩阵的矩阵来进行投影操作，其效果与直接由 X 矩阵构成的投影矩阵的效果一样。下面还将看到这一性质可直接拓广至对双线性矩阵进行投影的情况。

设 $X=P\xi Q^t$，且 $X=USV^t$，

则有

$$UU^tXVV^t=X \tag{6-190}$$

这是因为，$UU^tX=UU^tP\xi Q^t=P\xi Q^t$，在此利用了 $UU^tP=P$，实因 U 与 P 构成的是同一个线性空间；同理，$XVV^t=P\xi Q^tVV^t=P\xi Q^t$，这是因为 V 与 Q^t 构成的是同一个线性空间。结合以上讨论的两点，可得

$$UU^tXVV^t=(UU^tP)\xi(Q^tVV^t)=P\xi Q^t=X \tag{6-191}$$

广义秩消失因子分析法提出后，被广泛地应用于二维荧光光谱数据和液相色谱与光谱联用数据的定量分析[118~121]。

（三）残差双线性分解法

【基本思路】

残差双线性分解法由 Ohman，Geladi 和 Wold 共同提出[122~123]，其基本思路仍是利用双线性纯组分矩阵秩为一的基本原理，先是把含未知干扰的样本矩阵按式(6-146)所示那样分成两部分，一部分为待测的分析物(X_i，$i=1,2,\cdots,n$)，另一部分为未知干扰背景 R，继交替对数据进行两步分析，首先在固定背景的基础上对分析物进行校正；然后固定得到校正的分析物相对浓度，对干扰背景进行主成分分解，经不断交替进行上述两个过程，直至背景干扰的分解可在已知主成分数的基础上完全分解，且校正的分析物相对浓度达到收敛。这一算法的提出，为灰色分析体系的直接解析又提供了一种新思路，在本节将对这一算法给出详细的介绍。

对于式(6-146)所示的灰色分析体系数学模型，如直接采用第二节所述方法进行校正，由于干扰背景矩阵 R 的存在，所得校正结果必然是不佳的，然而如果利用双线性纯组分矩阵秩为一的条件，经过对样本矩阵的秩分析，就可预先知道该含未知干扰的样本矩阵中含有多少物种，从而也可算得在 R 矩阵中所含的物种数。R 矩阵中所含的物种数（或称 R 矩阵的主成分数），是一个非常重要的信息，也就是有了这样的一个重要信息，才使得灰色分析体系的唯一解成为可能。残差双线性分解法的目标就在于，在给定的含未知干扰样本矩阵 Y，校正矩阵 X_i（$i=1$，$\cdots$，n），以及背景干扰 R 矩阵中所含的物种数 m 的前提下，来正确估计式中的 c_i（$i=1$，$\cdots$，n）。

【基本算法】

残差双线性分解的基本算法，按以下步骤进行。

(1) 直接用 $X_i(i=1, \cdots, n)$ 对样本矩阵 Y 进行校正，得 $c_i^{(0)}(i=1, \cdots, n)$

(2) 对 $(Y-c_1^{(0)}X_1+c_2^{(0)}X_2+, \cdots, +c_n^{(0)}X_n)$ 进行主成分分解，并取前 m 个主成分构成背景干扰矩阵，即 $R^{(1)}=U^{(1)}V^{(1)t}$，在此 $U^{(1)}$ 和 $V^{(1)t}$ 为主成分分解前 m 个因子得分与因子荷载。

(3) 对于 $k=1, 2, \cdots\cdots$，进行下述迭代

① 计算 $(Y-U^{(k)}V^{(k)t})$，并用 X_i $(i=1, \cdots, n)$ 来对 $(Y-U^{(k)}V^{(k)t})$ 进行校正，得 $c_i^{(k)}$ $(i=1, \cdots, n)$；

② 对 $(Y-c_1^{(k)}X_1+c_2^{(k)}X_2+, \cdots, +c_n^{(k)}X_n)$ 进行主成分分解，并取前 m 个主成分构成背景干扰矩阵，即 $R^{(k+1)}=U^{(k+1)}V^{(k+1)t}$，在此 $U^{(k+1)}$ 和 $V^{(k+1)t}$ 为主成分分解前 m 个因子得分与因子荷载。

③ 计算 $(Y-c_1^{(k)}X_1+c_2^{(k)}X_2+, \cdots, +c_n^{(k)}X_n-U^{(k+1)}V^{(k+1)t})$ 的方差，如果它们大于量测误差方差则转向①继续迭代，否则，停止迭代。

【讨论】

以上算法只能说是残差双线性分解法的基本算法，实因其原始算法中的一个加速过程在本节中没有给出介绍。这是因为本书作者认为该加速过程虽有一定功效，但却有失烦琐和干扰对主要算法的理解，所以在此省略，读者可直接参阅文献[122]。

另外，对于第一步和迭代过程中的第①步，我们也没有给出具体的方法，原则上，就可简单地采用第三章中介绍的直接校正方法，即 MLR 方法，来计算就可以了。当然用 PLS 方法也是可行的，在残差双线性分解法原始文献中就是采用 PLS 来进行校正的，有兴趣的读者可直接参阅文献[122]。

Ohman，Geladi 和 Wold 将上述讨论的残差双线性分解法用于联用色谱（HPLC-DAD）数据的解析，并与广义秩消失因子分析法进行了比较研究，所得的结果示于表 6-11。

表 6-11 残差双线性分解法与广义秩消失因子分析法解析结果的预测误差比较表

序号	样本组成①				RBL		GRAFA	
	A	B	C	D	A	B	A	B
1	2	2	0	0	0.053	0.036	0.049	0.460
2	2	2	0	0	0.022	0.030	−1.253	−1.435
3	2	2	0	0	−0.008	−0.003	−0.722	0.267
4	2	2	0	0	0.020	0.064	0.022	−1.077
5	2	0	0	0	−0.313	0.007	0.110	−0.145
6	2	0	2	0	−0.030	0.007	−0.082	−0.395
7	2	0	0	2	0.122	−0.145	0.165	−0.180
8	2	0	2	2	0.147	0.090	0.295	0.907
9	0	2	0	0	−0.018	0.057	−0.015	0.095
10	0	2	2	0	−0.004	0.064	−0.020	0.185
11	0	2	0	2	−0.002	−0.116	−0.022	−0.152
12	0	2	2	2	0.003	0.046	−0.202	−0.040
13	2	2	2	0	−0.012	−0.068	−0.337	3.587
14	2	2	0	2	−0.255	−0.131	−0.255	−0.852
15	2	2	2	2	−0.169	−0.339	−0.605	−0.690
16	1	0	3	3	−0.043	0.099	−0.150	−0.325
17	0	1	3	3	0.041	0.178	−0.170	0.0
18	3	0	1	1	−0.059	−0.744	0.045	−0.115

续表

序号	样本组成①				RBL		GRAFA	
	A	B	C	D	A	B	A	B
19	0	3	1	1	1.53	−10.17	0.045	2.017
20	2	2	2	2	−0.196	−0.411	−0.082	0.372
21	2	2	2	2	−0.060	0.047	−0.517	−1.165
22	2	2	2	2	−0.027	0.284	−0.119	−0.765
23	0	0	2	0	−0.075	−0.100	−0.060	−0.017
24	0	0	0	2	−0.014	0.036	−0.245	−0.127

① 其中组分 A 和 B 为分析物；C 和 D 为背景干扰；预测误差＝估计值－实际值；

A—二苯胺；B—二苯酮；C—联苯酰；D—丁二酸二酯。

在残差双线性分解法的原始文献中，该文的作者还对算法中的第三步的迭代过程的收敛给出了一个证明，因本书作者认为该证明将对此法的理解很有帮助，所以，也将在下面给出介绍。

在第三步的迭代过程中，我们不断地更新 $c_i^{(k)}$ $(i=1, \cdots, n)$、$\boldsymbol{U}^{(k)}$ 和 $\boldsymbol{V}^{(k)t}$，每一次更新就相当于一次新的逼近，这样就有

$$\|\boldsymbol{Y}-\sum c_i^{(k)}\boldsymbol{X}_i-\boldsymbol{U}^{(k+1)}\boldsymbol{V}^{(k+1)t}\| \leqslant \|\boldsymbol{Y}-\sum c_i^{(k-1)}\boldsymbol{X}_i-\boldsymbol{U}^{(k)}\boldsymbol{V}^{(k)t}\| \tag{6-192}$$

在此给出证明如下：

因为 $c_i^{(k)}(i=1,\cdots,n)$是从对$(\boldsymbol{Y}-\boldsymbol{U}^{(k)}\boldsymbol{V}^{(k)t})$进行基于最小二乘最优意义的校正所得，(在这一点上，MLR 和 PLS 是等同的)，所以有

$$\|\boldsymbol{Y}-\sum c_i^{(k)}\boldsymbol{X}_i-\boldsymbol{U}^{(k)}\boldsymbol{V}^{(k)t}\| \leqslant \|\boldsymbol{Y}-\sum t_i\boldsymbol{X}_i-\boldsymbol{U}^{(k)}\boldsymbol{V}^{(k)t}\| \tag{6-193}$$

式中，t_i $(i=1, \cdots, n)$ 为任一其他可能非最小二乘意义的估计或包括 $c_i^{(k-1)}$ $(i=1, \cdots, n)$ 在内的任意估计。另一方面，$\boldsymbol{U}^{(k+1)}$ 和 $\boldsymbol{V}^{(k+1)t}$ 为 $(\boldsymbol{Y}-\sum c_i^{(k)}\boldsymbol{X}_i)$ 的主成分分解所得的前 m 个因子得分与因子荷载，所以，从最小二乘意义上来说，(实因主成分分析亦为一基于最小二乘原理的方法)，有

$$\begin{aligned}\|\boldsymbol{Y}-\sum c_i^{(k)}\boldsymbol{X}_i-\boldsymbol{U}^{(k+1)}\boldsymbol{V}^{(k+1)t}\| &\leqslant \|\boldsymbol{Y}-\sum c_i^{(k)}\boldsymbol{X}_i-\boldsymbol{H}\boldsymbol{G}^t\| \\ &\leqslant \|\boldsymbol{Y}-\sum c_i^{(k)}\boldsymbol{X}_i-\boldsymbol{U}^{(k)}\boldsymbol{V}^{(k)t}\|\end{aligned} \tag{6-194}$$

因为式 (6-194) 中 $\boldsymbol{H}$ 和 $\boldsymbol{G}^t$ 为任意其他矩阵的前 m 列或前 m 行，当然也包括 $\boldsymbol{U}^{(k)}$ 和 $\boldsymbol{V}^{(k)t}$ 在内。

（四）约束背景双线性分解法

【基本思路】

从上一节的讨论可以看到，残差双线性分解法实质上相当于一个不断逼近过程，此法通过交替调整待测分析物的浓度估计值和干扰背景的主成分估计来不断逼近一个目标函数$(\boldsymbol{Y}-\sum c_i\boldsymbol{X}_i-\boldsymbol{U}\boldsymbol{V}^t)$，使其达到量测噪声水平。所以，残差双线性分解法有可能转化成一优化问题。

现在，来具体考察一下灰色分析体系的数学模型［式 (6-146)］：

$$\boldsymbol{Y}=c_1\boldsymbol{X}_1+c_2\boldsymbol{X}_2+, \cdots, +c_n\boldsymbol{X}_n+\boldsymbol{R}+\boldsymbol{E}$$

从式 (6-146) 可以看到，这里已知量是样本量测矩阵 $\boldsymbol{Y}$ 和校正矩阵 $\boldsymbol{X}_i(i=1, \cdots, n)$；另一方面，似还可得到一个隐含在纯组分量测矩阵都是双线性矩阵之假设下的一个已知量。可以预先求出 $\boldsymbol{Y}$ 矩阵的秩，继用求得的 $\boldsymbol{Y}$ 矩阵的秩减去待测组分数目，又可得未知干扰背景矩阵的秩，从而也就得到了 $\boldsymbol{R}$ 矩阵中所含未知干扰的物种数。注意到这一信息是一非常重要的信息，也就是利用了这一信息，才使双线性二维数据的灰色分析体系的解析可得到有物理意义的唯

一解。

利用样本量测矩阵 $\boldsymbol{Y}$、校正矩阵 $\boldsymbol{X}_i$（$i=1, \cdots, n$）和未知干扰背景矩阵的秩这些已知条件，梁逸曾、Manne 和 Kvalheim 将灰色分析体系的解析问题转化成为一个优化问题，提出了约束背景双线性分解法[124]。现将该法的基本思路介绍如下。

将灰色分析体系的数学模型改写成为

$$\boldsymbol{Y}=\sum c_i\boldsymbol{X}_i+\boldsymbol{R}+\boldsymbol{E}$$
$$=\sum c_i\boldsymbol{X}_i+\sum \boldsymbol{u}_j\boldsymbol{v}_j^{\prime}+\boldsymbol{E} \quad (i=1, \cdots, n;\ j=1, \cdots, m) \tag{6-195}$$

式中，$\boldsymbol{u}_j$ 和 $\boldsymbol{v}_j^{\prime}$（$j=1, \cdots, m$）分别为对 $\boldsymbol{R}$ 矩阵进行主成分分析所得的因子得分和因子载荷。从式（6-195）可以看出，假设先估计 c_i，如果 c_i（$i=1, \cdots, n$）的估计不正确，过高或过低，都将使（$\boldsymbol{Y}-\sum c_i\boldsymbol{X}_i$）中还包含了校正矩阵 $\boldsymbol{X}_i$（$i=1, \cdots, n$）的信息，这就导致不可能只用 m 个主因子对 $\boldsymbol{R}$ 矩阵，亦即（$\boldsymbol{Y}-\sum c_i\boldsymbol{X}_i$）矩阵进行完全分解，也就是说，此时（$\boldsymbol{Y}-\sum c_i\boldsymbol{X}_i-\sum \boldsymbol{u}_j\boldsymbol{v}_j^{\prime}$）中还包含了未扣尽的 $\boldsymbol{X}_i$（$i=1, \cdots, n$）的信息，当然，其方差也就大于量测误差了。当且仅当 c_i（$i=1, \cdots, n$）被正确估计时，才能用 m 个主因子对（$\boldsymbol{Y}-\sum c_i\boldsymbol{X}_i$）矩阵进行完全分解，也只有此时（$\boldsymbol{Y}-\sum c_i\boldsymbol{X}_i-\sum \boldsymbol{u}_j\boldsymbol{v}_j^{\prime}$）的方差才与量测误差相当。这样由式（6-195）所定义的灰色分析体系的定量校正问题就转化成为一个以 $\|\boldsymbol{Y}-\sum c_i\boldsymbol{X}_i-\sum \boldsymbol{u}_j\boldsymbol{v}_j^{\prime}\|^2$ 为目标函数的一个优化问题。即在给定的 $\boldsymbol{X}_i$（$i=1, \cdots, n$）和 m 的条件下，对 c_i（$i=1, \cdots, n$）的估计进行寻优，当目标函数 $\|\boldsymbol{Y}-\sum c_i\boldsymbol{X}_i-\sum \boldsymbol{u}_j\boldsymbol{v}_j^{\prime}\|^2$ 达到最小时，c_i（$i=1, \cdots, n$）的估计即为要寻找的待测物的相对浓度。

如果对 c_i（$i=1, \cdots, n$）估计的寻优需在（$-\infty, +\infty$）全空间进行搜索的话，这一最优化问题的求解仍不是一件易事。幸而在此要求的是相对浓度，它们不可能为负数。利用一些别的已知条件，可以把寻优搜索空间约束在一个（$0, a$）空间中进行，这也是为什么称之为约束背景双线性分解法的原因。

【基本算法】

1. c_i（$i=1, \cdots, n$）估计的约束界

因要求的是相对浓度值，所以，它们不可能为负数，即

$$c_i \geqslant 0 \quad (i=1, \cdots, n)$$

此为 $c_i(i=1, \cdots, n)$估计的下界。

另外，利用光谱量测值不为负的约束，即光谱的量测值 R_{kj}和 X_{ikj}不可能为负的条件，我们可以导出 c_i（$i=1, \cdots, n$）估计的上界。在此，R_{kj}和 X_{ikj}分别表示 $\boldsymbol{R}$ 矩阵和 X_i 矩阵中第 k 行第 j 列的元素。

设量测误差矩阵中的所有元素 E_{kj}（$k=1, \cdots, w$；$j=1, \cdots, r$）的绝对值都小于某一常数，即

$$|E_{kj}| \leqslant E$$

因为量测值 R_{kj}不可能为负，所以有

$$R_{kj}=Y_{kj}-\sum c_i X_{ikj} \geqslant -E \tag{6-196a}$$

或

$$\sum c_i X_{ikj} \leqslant Y_{kj}+E \tag{6-196b}$$

同时，因 c_i 和 X_{ikj}都不能为负的条件，

$$c_i=c_i X_{ikj}/X_{ikj} \leqslant \sum c_i X_{ikj}/X_{ikj} \leqslant (Y_{kj}+E)/X_{ikj} \tag{6-197}$$

或者可写为，对于所有的 $k=1, \cdots, w$；$j=1, \cdots, r$，

$$c_i \leqslant \mathrm{Max}(c_i) = \mathrm{Min}[(Y_{kj}+E)/X_{ikj}] \quad (i=1,\cdots,n) \tag{6-198}$$

将式 $c_i \geqslant 0$ 和式(6-198)连起来，就得到了浓度估计的约束界。

2. 背景干扰矩阵的秩估计

因为 $\boldsymbol{Y}$ 矩阵的秩减去待测组分数目就可得未知干扰背景矩阵的秩，即

$$m = \mathrm{rank}(\boldsymbol{R}) = \mathrm{rank}(\boldsymbol{Y}) - n \tag{6-199}$$

所以，正确估计 $\boldsymbol{Y}$ 矩阵的秩也是本法能否成功的一个关键。估计矩阵的秩的方法很多，但大多是基于主成分分析的方法来完成的，有关这一点将在第四节详细讨论。另外，如果 m 等于零，这说明 $\boldsymbol{R}$ 矩阵实际不存在，此时即可直接采用白色分析体系的解析方法来进行解析了。

3. 优化方法

在目标函数 $\|\boldsymbol{Y}-\sum c_i\boldsymbol{X}_i-\sum \boldsymbol{u}_j\boldsymbol{v}_j^t\|^2$ 中，$\boldsymbol{u}_j$ 和 $\boldsymbol{v}_j^t$ 也是 c_i（$i=1,\cdots,n$）的函数，因 $\boldsymbol{u}_j$ 和 $\boldsymbol{v}_j^t$ 也是主成分分解的因子得分和因子载荷，对目标函数直接求导几乎不可能，所以，需采用毋需依靠求导的优化方法，如单纯形优化方法[125]或 Powell 方法[126,127]。因此问题是一个约束优化问题，如采用单纯形优化方法须在每步都计算一次边界条件，所以在文献[124]中是采用 Powell 方法来进行的。具体算法步骤如下：

① 选择一个落在约束区间的初始估计浓度矢量点 $\boldsymbol{c}^{1,0}=\{c_1^0, c_2^0, \cdots, c_n^0\}$ 和任一组张成约束区间的正交基矢量 $\{\boldsymbol{v}^{k,1}, \boldsymbol{v}^{k,2}, \cdots, \boldsymbol{v}^{k,n}\}$。

② $k=1, 2, 3, \cdots$

③ i 从 1 到 n，计算

$$\boldsymbol{c}^{k,i} = \boldsymbol{c}^{k,i} + \theta^{k,i}\boldsymbol{v}^{k,i} \tag{6-200}$$

在此 $\theta^{k,i}$ 使下列包含 θ 的函数取得极小值，

$$\phi(\theta) = f(\boldsymbol{c}^{k,i-1}+\theta\boldsymbol{v}^{k,i}, \boldsymbol{u}_j, \boldsymbol{v}_j^t) \tag{6-201}$$

④ 令 $\boldsymbol{v}^{k,n+1}=(\boldsymbol{c}^{k,n}-\boldsymbol{c}^{k,i})/\|\boldsymbol{c}^{k,n}-\boldsymbol{c}^{k,i}\|$，并且计算

$$\boldsymbol{c}^{k,n+1} = \boldsymbol{c}^{k,n} + \theta^{k,n+1}\boldsymbol{v}^{k,n+1}$$

在此 $\theta^{k,n+1}$ 使下列包含 θ 的函数取得极小值，

$$\phi(\theta) = f(\boldsymbol{c}^{k,n}+\theta\boldsymbol{v}^{k,n+1}, \boldsymbol{u}_j, \boldsymbol{v}_j^t)$$

⑤ i 从 1 到 n，计算

$$\Delta_\pi = \max[f(\boldsymbol{c}^{k,i-1}, \boldsymbol{u}_j, \boldsymbol{v}_j^t) - f(\boldsymbol{c}^{k,i}, \boldsymbol{u}_j, \boldsymbol{v}_j^t)]$$

在此，下标 π 表示 Δ 是在第 π 步获得的极大值，$1\leqslant\pi\leqslant n$。

⑥ 如果

$$|\theta^{k,n+1}| \leqslant \{[f(\boldsymbol{c}^{k,0}, \boldsymbol{u}_j, \boldsymbol{v}_j^t) - f(\boldsymbol{c}^{k,n+1}, \boldsymbol{u}_j, \boldsymbol{v}_j^t)]/\Delta_\pi\}^{1/2}$$

那么，i 从 1 到 n，令

$$\boldsymbol{v}^{k+1,i} = \boldsymbol{v}^{k,i}$$

并转向第⑦步，否则令

$$\boldsymbol{v}^{k+1,i} = \boldsymbol{v}^{k,i} \quad (i=1,2,\cdots,\pi-1)$$

$$\boldsymbol{v}^{k+1,i} = \boldsymbol{v}^{k,i+1} \quad (i=\pi,\pi+1,\cdots,n-1)$$

$$\boldsymbol{v}^{k+1,n} = \boldsymbol{v}^{k,n+1}$$

转向第⑦步

⑦ 如果

$$f(\boldsymbol{c}^{k,i}, \boldsymbol{u}_j, \boldsymbol{v}_j^t)/(w\times r-n) \leqslant 2\times E$$

或者是，对于 $i=1,2,\cdots,n$，都有

$$c_i^{k+1}-c_i^k\leqslant 0.0005$$

则停止计算，否则令 $k=k+1$，转向第③步。

由上式给出的 0.0005 是一个终止指标，即如果前后两次迭代的浓度差值小于 0.0005 时则认为已达到要求，这是因为我们认为紫外可见光谱最多能达到量测精度，对于不同的分析仪器，应该有不同的终止指标。该算法的第④步到第⑥步是保证搜索方向 $\{v^{k,1}, v^{k,2}, \cdots, v^{k,n}\}$ 总能线性无关[125]，另外，在上述算法的第③和第④步的一维搜索优化算法在本文是采用了 0.618 法，或称黄金分割法来进行的。

4. 优化搜索区间的确定

上述讨论的 Powell 方法实质上是一种无约束优化方法，在优化过程中，只给出了优化的方向而没有给出优化搜索的区间，在此将对优化的搜索区间的确定进行必要的讨论。

对于给定的初始矢量 $\boldsymbol{c}^0$ 和寻优搜索方向 $\boldsymbol{v}$，由此可以构成一个直线方程

$$\boldsymbol{c}^0+\beta\boldsymbol{v}=\boldsymbol{c} \tag{6-202}$$

式中，β 就是我们要确定的一个实数，确定了 β 的变化区间也就等于确定了优化的搜索区间。实际上，搜索区间是由式（6-203）决定的。

$$0\leqslant c_i\leqslant \max(c_i)\quad (i-1,2,\cdots,n) \tag{6-203}$$

将式(6-202)代入式(6-203)，则有

$$0\leqslant c_i^0+\beta v_i\leqslant \max(c_i)\quad (i=1,2,\cdots,n) \tag{6-204}$$

这样，对于 $\boldsymbol{c}$ 矢量的每一个元素，β 都可以找到一对边界。现分三种不同情况加以讨论：

① 当 v_i 大于零时，

$$-c_i^0/v_i\leqslant\beta\leqslant(\max(c_i)-c_i^0)/v_i\quad (i=1,2,\cdots,n) \tag{6-205a}$$

② 当 v_i 小于零时，

$$-c_i^0/v_i\geqslant\beta\geqslant(\max(c_i)-c_i^0)/v_i\quad (i=1,2,\cdots,n) \tag{6-205b}$$

③ 当 v_i 等于零时，

$$0\leqslant\beta\leqslant\max(c_i)\quad (i=1,2,\cdots,n) \tag{6-205c}$$

由上述式(6-205)确定的是一组边界，对于下界应取其中的最大值，但对于上界则应取其最小值。确定寻优搜索边界的几何意义示于图 6-18。

在这里的初始矢量 $\boldsymbol{c}^0$ 是一个具有相对意义的初始矢量 $\boldsymbol{c}^0$，即每一个方向所得到的最优点实际上就是下一次搜索中的初始矢量。

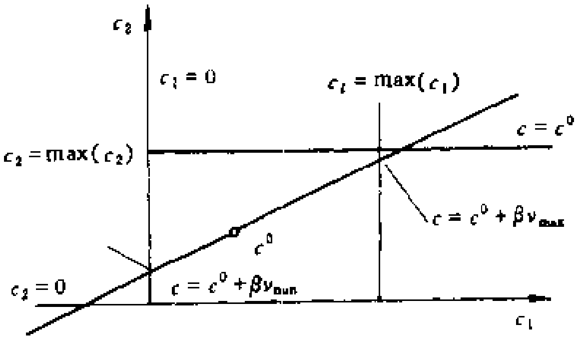


图 6-18 确定寻优搜索边界的几何示意图

β_{min}为下界点，β_{max}为上界点

【讨论】

约束背景双线性分解法实质上是把灰色分析体系的校正问题转换成为一个约束优化问题。前一节讨论的残差双线性分解法实质上也相当于一个不断优化逼近过程，不同的是，该法是通过交替调整待测分析物的浓度估计值和干扰背景的主成分估计来不断逼近的，而且，此逼近过程的初始点就是不考虑背景干扰的简单对校正矩阵的回归结果，可以想象，如果背景干扰严重时，此点离真实最优点是很远的，如果在交替逼近过程中存在着局部最优点的话，就有可能落入局部最优的危险，这一点从表 6-11 所列的结果似乎也可以看出。残差双线性分解

法的结果除少数几个外一般都比较好，只有第 32 号为一非常坏的结果，这就有可能是落入了局部最优，不然，此结果无法解释，由式（6-194）给出的逼近过程总是会使目标函数（$Y-\sum c_iX_i-UV'$）不断变小的，所以此解只可能是局部最优解，即虽然此解离真实解很远，当目标函数（$Y-\sum c_iX_i-UV'$）的值在该领域可能却是最小的。

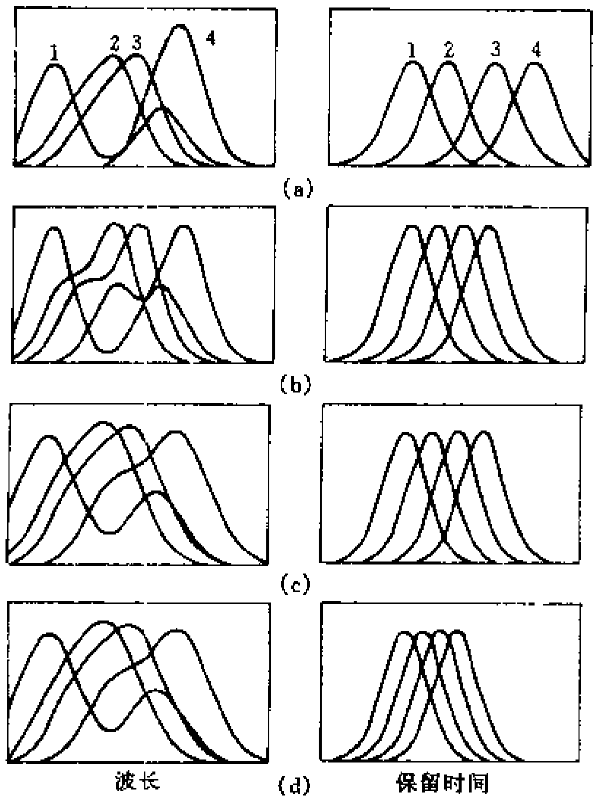


图 6-19　模拟体系的纯组分色谱和光谱图

为探讨残差双线性分解法的局部最优解的情况，我们对其进行了计算机模拟实验。图 6-19 示出了模拟双线性体系的色谱和光谱图。表 6-12 给出了两种方法的计算结果进一步证实这一结论，我们算出了（a）（b）（c）（d）四个体系的响应曲面（参见图 6-20）。从图 6-20 可以看出，此响应曲面类似鞍形，如初始点远离真实解时，是有可能落入局部最优的。

虽然约束背景双线性分解法是用浓度估计的约束来减小落入局部最优的危险，但是，因该法在此使用的 Powell 算法仍是一个局部最优方法，所以还不能保证对任何情况都不落入局部最优。随着化学计量学的深入研究，近年来出现了几个全局最优算法，著名的有模拟退火[128,129]和遗传算法[130,131]。谢玉珑等用模拟退火方法研究了约束背景双线性分解法的局部最优情况[132]，最近，陈文灿等也用遗传算法对此进行了类似研究[133]，均得到满意结果。

表 6-12　图 6-19 所示模拟体系的残差双线性分解法（RBL）与约束背景双线性分解法（CBBL）结果比较①

分析体系	分析物	未知干扰	RBL 估计		CBBL 估计	
a	1，2	3，4	0.9998	1.002	0.9999	1.00004
	1，3	2，4	0.9995	1.002	1.0001	1.0003
	2，3	1，4	1.0003	1.0003	0.99996	0.99997
	1，4	2，3	1.0004	1.0003	1.0000	0.99999
b	1，2	3，4	0.9999	1.003	0.99997	1.0000
	1，3	2，4	0.9995	1.004	1.0000	1.0000
	2，3	1，4	0.9999	1.015	0.99997	1.0000
	1，4	2，3	0.9982	1.003	1.0000	1.0000
c	1，2	3，4	0.9998	1.002	0.99999	1.00004
	1，3	2，4	2.026	1.041	1.0001	1.003
	2，3	1，4	0.9993	1.008	0.99996	0.99997
	3，4	1，2	2.818	1.1002	1.0000	0.99999
d	1，2	3，4	1.006	1.925	1.0000	1.0003
	1，3	2，4	0.9869	1.138	1.0000	1.0003
	2，3	1，4	1.024	0.9835	0.99996	0.99997
	1，4	2，3	1.558	1.090	1.0000	1.00004
	2，4	1，3	1.146	0.9635	1.0002	0.99992
	3，4	1，2	2.088	1.111	0.9997	1.00008

① 表中的体系和组分编号皆由图 6-19 给出。

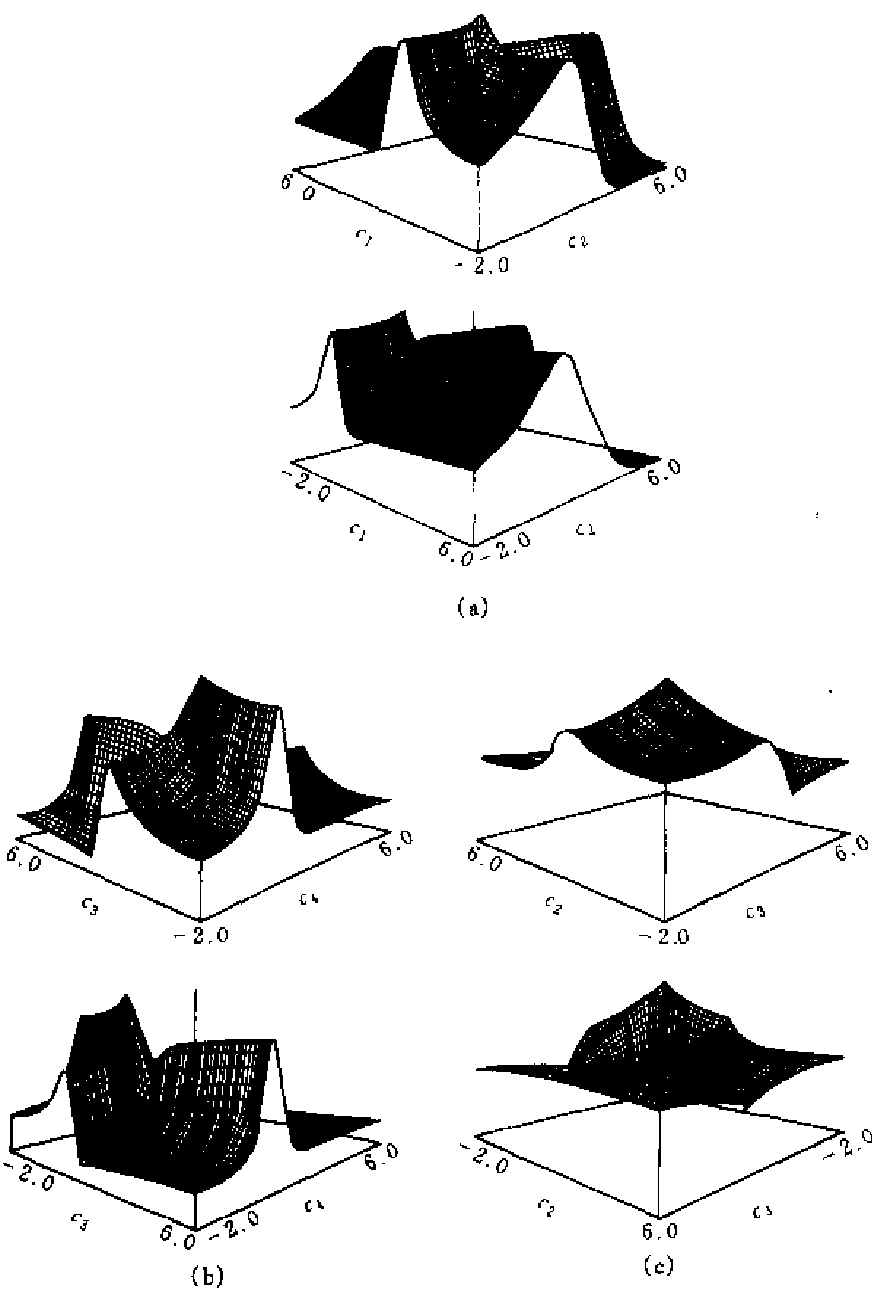


图 6-20　几个模拟体系的响应曲面图

从本章对灰色分析体系的矩阵校正的分析方法的讨论中可以看到，如果采用矩阵数据，从数学意义上来说，是可以得到唯一解，可是，在实际应用中还存在一些困难。首先，在灰色分析体系的矩阵校正的方法中，实际上都隐含着一个假设，就是能准确地获得任一矩阵的秩的正确估计，然而，在实际数据中实现这一点却很不容易，尤其是当样本中的未知背景干扰比较复杂时更是如此，实因在矩阵秩估计中，影响因素很多，如纯组分波谱的重叠程度，组分数目，纯组分之间浓度的相对比例，噪声的形状（如等方差噪声与非等方差噪声）等都直接影响矩阵秩的正确估计，从这一角度来说，本章所讨论的灰色分析体系的校正方法对于过于复杂的体系直接解析还存在一些问题，不能认为上述方法对于任一灰色分析体系均可得到很准确的唯一解，是随样本的复杂程度而变化的。其次，对于上述方法应用于分析化学的实际问题解决中，此类方法还存在的另一类困难是来源于实际分析仪器所产生数据的局限，对于二维荧光光谱仪所产生的二维双线性谱一般不存在问题，但是，对于另一类可产生二维双

线性数据的色谱波谱联用的分析仪器，则因色谱的条件的难于控制，而使色谱在保留时间的方面难于保证准确无误的重现性，所以也给上述讨论的方法带来困难。如何根据实际情况来对联用仪器所产生的二维数据进行适当的预处理，使之合乎灰色分析体系矩阵校正方法的基本要求和假设，仍是化学计量学家们需要继续研究的一个重要课题。

第四节　黑色分析体系的多元分辨方法

对于分析试样毫无验前信息，即有关其物种数，哪几种化学物种及其浓度范围皆不清楚，分析化学的任务是首先确定其物种数，进而解析出各纯物种的谱图（可以是光谱、波谱，等），即将其首先转化成为白色分析体系，然后进行定量分析。这类分析体系是分析化学中最难的一类体系，因其像个黑匣子，故我们称之为“黑色分析体系”。在以后的讨论中，我们将看到分析化学计量学家们已研究出了不少基于矩阵数据结构和张量数据结构的化学计量学解析方法，尤其是对于色谱联用仪器，如液相色谱与二极管阵列联用仪（HPLC-DAD），气相色谱与质谱联用仪（GC-MS）、气相色谱与红外光谱联用仪（GC-IR）等所产生的矩阵数据，已发展了一系列解析新方法，可望解决黑色分析体系的快速定性定量分析问题。

黑色分析体系的分辨算法在本章将以矩阵分辨算法和张量分辨算法为线索展开。从理论上说，基于矩阵分辨的方法很难得到有物理意义的唯一解。但是，如果存在一些具有特殊条件的数据，其中特别是色谱联用仪器所产生的二维数据，情况就发生了变化。这些仪器包括液相色谱与二极管阵列联用仪（HPLC-DAD），气相色谱与质谱联用仪（GC-MS）、气相色谱与红外光谱联用仪（GC-IR）等。因这些数据不但同时包含了光谱和色谱的信息，而且，色谱还具备很强的分离特性，可使一个很复杂的混合体系分解成为相对简单的子系统，并且这些子系统还同时具有一些别的有用特征，这就为分析化学计量学家们提供了机会。近几年来，基于自模式曲线分辨法，分析化学计量学家们提出了很多的新算法，对于一般的色谱重叠峰，均可给出令人满意的结果，如渐进因子分析法、窗口因子分析法、直观推导式演进特征投影法、正交投影分辨法、子窗口因子分析法等。对于这些方法及其发展历史，本书将给出较详细的介绍和讨论，为本书的一个重点。对于张量分辨方法本书也将作出必要的介绍。

一、基于主成分分析的体系组分数确定方法

借主成分分析确定体系独立组分数的有关理论和方法，是黑色分析体系解析的至关重要的第一步。多年来有关此方面已有较多研究[134~139]，Malinowski 对此进行了全面系统的研究，提出了因子分析（主成分分析）的误差理论[140]，并给出了几种确定化学混合体系物种数的判别标准。

所谓主成分分析（PCA），又称因子分析（FA），这两种方法在数学上就是对数据量测矩阵的协方差阵进行正交变换，找到量测矩阵的协方差阵的特征值和特征向量（参见第十一章），继而确定体系的主成分或主因子，即化学混合体系的纯物种数。

设 $\boldsymbol{X}$ 为化合物光谱的量测矩阵。其中 x_{ij}为第 i 个混合物在第 j 个波长（或更一般地称为敏感通道 j）的量测值，在此对混合物样本有一限制，即每个混合物中各组分的相对浓度应不尽相同或对应成比例，用矩阵可具体表示为

$$\boldsymbol{X}=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{6-206}$$

式中，m 为混合物波谱数目，n 为波长（分析通道）数目。一般要求 m 和 n 都大于体系实际存在的独立组分数（即体系的纯物种数）。

显然，如不存在量测误差，则 X 矩阵的秩就代表体系中存在的独立组分数，这是因为矩阵的秩就代表此矩阵中线性无关的向量的个数。确定矩阵秩的方法有几种，本文主要介绍主成分分析的方法。

先求出量测矩阵 X 的协方差阵 Z，

$$Z=X^tX \tag{6-207}$$

协方差阵 Z 为对称正定阵，这样就可采用正交变换的方法来求出它的特征值和特征向量，不为零的特征值的个数就是矩阵的秩。如设 X^* 为不含任何误差的数据矩阵，则有

$$Q^{*t}X^{*t}X^*Q^*=\begin{bmatrix} \lambda_1^* & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2^* & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_d^* & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{6-208}$$

式中，d 为独立组分数，且 $\lambda_i^*>\lambda_{i+1}^*$，$Q^*$ 为 X^* 所对应的正交特征向量矩阵。然而，实际上任何量测都存在误差，即

$$X=X^*+E \tag{6-209}$$

式中，E 为误差矩阵。如果直接对量测矩阵 X 的协方差阵 Z 进行正交分解，一般可得

$$Q^tX^tXQ=\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_d & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \lambda_n \end{bmatrix} \tag{6-210}$$

原先那些为零的特征值因误差的影响都转变成非零，仍然有 $\lambda_i>\lambda_{i+1}$。这样，探讨体系中的独立组分数（主因子数）的问题，就转换成在误差的影响之下，(6-208) 式右边对角阵中原为零的特征值将会怎样转化为非零，以及如何来确定存在误差的矩阵 X 的独立组分数的问题。

（一）误差**扰动下的协方差阵特征值变化限制**

由式 (6-207)，可求得

$$\begin{aligned} Z &= X^tX=(X^*+E)^t(X^*+E) \\ &= X^{*t}X^*+X^{*t}E+E^tX^*+E^tE \\ &= Z^*+\delta Z \end{aligned}$$

Z^* 在 δZ 扰动的影响下，其特征值 λ_i^* 会发生什么样的变化，变化的幅度受什么限制？在此，可采用形式求变分的方法，并略去二阶以上的小量，即可求出特征值 λ_i^* $(i=1,2,\cdots,d)$ 的变化情况。根据特征值与特征向量的性质，(6-208)式可写成如下形式

$$Z^*q_j^*=\lambda_j^*q_j^* \tag{6-211}$$

式中，q_j^* 为矩阵 Q^* 中第 j 个列向量，λ_j^* 为其相对应特征向量的特征值，当 $j>d$ 时，皆为零。对式（6-211）微分得

$$\delta(Z^*)q_j^*+Z^*\delta q_j^*=\lambda_j^*\delta(q_j^*)+\delta(\lambda_j^*)q_j^* \tag{6-212}$$

两边对相应特征向量 q_j^* 作内积，

$$q_j^{*\prime}\delta(Z^*)q_j^*+q_j^{*\prime}Z^*(\delta q_j^*)=\lambda_j^*q_j^{*\prime}\delta(q_j^*)+\delta(\lambda_j^*)q_j^{*\prime}q_j^* \tag{6-213}$$

由于 Z^* 为一对称阵，即有

$$q_j^{*\prime}Z^*=(\lambda_j^*q_j^*)'=\lambda_j^*q_j^{*\prime} \tag{6-214}$$

这样，就可将式(6-213)左边的第二项和右边的第一项略去，所以有

$$q_j^{*\prime}\delta(Z^*)q_j^*=\delta(\lambda_j^*)q_j^{*\prime}q_j^*$$

即得

$$\delta(\lambda_j^*)=[q_j^{*\prime}\delta(Z^*)q_j^*]/(q_j^{*\prime}q_j^*) \tag{6-215}$$

对式(6-215)取模得

$$|\delta(\lambda_j^*)|=\|q_j^*\|\|\delta(Z^*)\|\|q_j^*\|/(\|q_j^*\|^2) \tag{6-216}$$

式中，$\|\cdot\|$ 表示任意一种矩阵范数。

因 Q^* 为一正交矩阵，所以有 $\|q_j^*\|^2=1$，亦即 $\|q_j^*\|=1$，于是，式(6-216)可简化为

$$|\delta(\lambda_j^*)|\leqslant\|\delta(Z^*)\| \tag{6-217}$$

从式(6-217)可以看出，所有实对称矩阵的特征值问题是良态的，它仅与 $\delta(Z^*)$ 的范数有关，而主成分分析中所遇到的数据协方差阵 Z 均系实对称矩阵，这就为借主成分分析来确定体系独立组分数提供了理论依据。

仍从式（6-217）可以看出，$|\delta(\lambda_j^*)|$ 小于 $\|\delta(Z^*)\|$，如采用极大范数，则 $|\delta(\lambda_j^*)|$ 要小于 $\delta(Z^*)$ 矩阵的最大特征值，这提示我们如按（$X^{*\prime}E+E'X^*+E'E$）构成 $\delta(Z^*)$ 矩阵，再求出它的特征值，即可得出 $|\delta(\lambda_j^*)|$ 的上限，据此即可判断体系中至少含有多少个独立组分数。

（二）因子分析的误差理论

由于实验误差，任何数据量测矩阵皆可表示为两部分，即

$$x_{ij}=x_{ij}^*+e_{ij} \tag{6-218}$$

实际上，式（6-218）也就是式（6-209）的标量式。另外，根据 $Q'X'XQ$，并令 $U=XQ$，则有

$$X=UQ'$$

在此利用了 Q 矩阵是正交矩阵的性质。根据矩阵乘法法则可得

$$x_{ij}=\sum_{k=1}^{n}u_{ik}q_{kj} \tag{6-219}$$

而根据 $Q^{*\prime}X^{*\prime}X^*Q^*$ 式，并令 $X^*=U^*Q^{*\prime}$，则有

$$x_{ij}^*=\sum_{k=1}^{d}u_{ik}^*q_{kj}^* \tag{6-220}$$

式（6-219）、(6-220）说明，X 可由 n 个相互正交的向量 q_j（$j=1，2，\cdots，n$）线性表出，而 X^* 则可由 d 个相互正交的向量 q_j^*（$j=1，2，\cdots，d$）线性表示，因误差阵 E 包括在量测矩阵 X 之中，所以有

$$e_{ij}=\sum_{k=1}^{d}\sigma_{ik}^1q_{kj}+\sum_{k=d+1}^{n}\sigma_{ik}^0q_{kj} \tag{6-221}$$

即，误差矩阵 E 也可以根据量测矩阵 X 所求得的正交矩阵 Q 来线性表出，在此，将其分成两部分，第一部分 $\sum\sigma_{ik}^1q_{kj}$，它包含前 d 个大特征值所对应的特征向量 q_j（$j=1,2,\cdots,d$），Malinowski

将这 d 个特征向量称为主轴；第二部分为 $\sum \sigma_{ik}^0 q_{kj}$，属于纯粹由量测误差而引出的那部分特征向量 $\boldsymbol{q}_j(j=d+1,d+2,\cdots,n)$，一般说来它们所对应的特征值都很小，Malinowski 将它们称为次轴。

将(6-218)式用(6-220)式和(6-221)式来表示，可得

$$x_{ij}=\sum_{k=1}^{d}(u_{ik}^* q_{kj}^* + \sigma_{ik}^1 q_{kj}) + \sum_{k=d+1}^{n} \sigma_{ik}^0 q_{kj} \tag{6-222}$$

对照(6-219)式，令 $u_{ij}=u_{ik}^* q_{kj}^*/q_{kj}+\sigma_{ik}^+$，那么，式(6-222)可写成

$$x_{ij}=\sum_{k=1}^{d}(u_{ik}q_{kj}) + \sum_{k=d+1}^{n} \sigma_{ik}^0 q_{kj} \tag{6-223}$$

如写成矩阵的形式，则有

$$\boldsymbol{X}=\boldsymbol{U}^*\boldsymbol{Q}^t+\Delta\boldsymbol{Q}^t \tag{6-224}$$

式(6-224)中的 $\boldsymbol{Q}$ 即为式 $\boldsymbol{X}=\boldsymbol{U}\boldsymbol{Q}^t$ 中的特征向量阵，而

$$\boldsymbol{U}^*=\begin{bmatrix} u_{11}^* & \cdots & u_{1d}^* & 0 & \cdots & 0 \\ u_{21}^* & \cdots & u_{2d}^* & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ u_{m1}^* & \cdots & u_{md}^* & 0 & \cdots & 0 \end{bmatrix} \tag{6-225}$$

$$\Delta=\begin{bmatrix} 0 & \cdots & 0 & \sigma_{1,d+1}^0 & \cdots & \sigma_{1,n}^0 \\ 0 & \cdots & 0 & \sigma_{2,d+1}^0 & \cdots & \sigma_{1,n}^0 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \sigma_{m,d+1}^0 & \cdots & \sigma_{m,n}^0 \end{bmatrix} \tag{6-226}$$

式(6-224)右边分成了两部分，即

$$\boldsymbol{X}=\boldsymbol{X}^+ + \boldsymbol{X}^0$$

其中

$$\boldsymbol{X}^+=\boldsymbol{U}^*\boldsymbol{Q}^t \tag{6-227a}$$

$$\boldsymbol{X}^0=\Delta\boldsymbol{Q}^t \tag{6-227b}$$

在此，$\boldsymbol{X}^+$ 为主轴(即主因子)所构成的矩阵，它包含了量测矩阵 $\boldsymbol{X}$ 的主要信息，而 $\boldsymbol{X}^0$ 则为次轴所构成的矩阵，它主要反映了量测误差的信息。

根据正交变换不改变矩阵的迹的原理(参见第十一章)，依照式

$$\boldsymbol{Q}^t\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{Q}=\mathrm{diag}(\boldsymbol{\lambda}),$$

有

$$\mathrm{tr}(\boldsymbol{X}^t\boldsymbol{X})=\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ik}^2=\mathrm{tr}(\boldsymbol{Q}^t\boldsymbol{X}^t\boldsymbol{X}\boldsymbol{Q})=\mathrm{tr}(\boldsymbol{U}^t\boldsymbol{U})$$

$$=\mathrm{trace}\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \lambda_d & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \lambda_{d+1} & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \lambda_n \end{bmatrix}=\sum\lambda_i$$

这些特征值亦可分成两组，一组由主因子阵给出，而另一组则由误差引出，即

$$\begin{aligned}\operatorname{tr}(X^{+t}X^{+}) &= \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^{+2} = \operatorname{tr}(Q^tX^{+t}X^{+}Q)\\ &= \operatorname{tr}(Q^tQU^{*t}U^{*}Q^tQ) = \operatorname{tr}(U^{*t}U^{*})\\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} u_{ij}^{*2}\end{aligned}$$

$$= \operatorname{trace}\begin{bmatrix}\lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0\\ \vdots & \vdots & & \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & \lambda_d & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0\\ \vdots & \vdots & & \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0\end{bmatrix} = \sum_{i=1}^{d}\lambda_i$$

而

$$\begin{aligned}\operatorname{tr}(X^{0t}X^{0}) &= \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^{0\,2} = \operatorname{tr}(Q^tX^{0t}X^{0}Q)\\ &= \operatorname{tr}(Q^tQ\Delta^t\Delta Q^tQ) = \operatorname{tr}(\Delta^t\Delta)\\ &= \sum_{i=1}^{m}\sum_{j=d+1}^{n} \sigma_{ij}^{0\,2}\end{aligned}$$

$$= \operatorname{trace}\begin{bmatrix}0 & 0 & \cdots & 0 & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0\\ \vdots & \vdots & & \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0 & \lambda_{d+1} & \cdots & 0\\ \vdots & \vdots & & \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & 0 & 0 & \cdots & \lambda_n\end{bmatrix} = \sum_{i=d+1}^{n}\lambda_i \tag{6-228}$$

在上述推导中，多次运用到正交矩阵的正交特性和矩阵求迹的一些法则(参见第十一章)，而且还利用了式(6-227)表出的关系及式(6-225)和式(6-226)表示的矩阵的正交特性。以后，为示以区别，特在前 d 个特征值上标一标记“+”，即 $\lambda_i^{+}(j=1,2,\cdots,d)$，以表示它们是来自主因子阵；而在后$(n\sim d)$个特征值上标以“0”，即记为 $\lambda_i^{0}(j=1,2,\cdots,d)$，来表示它们皆来自误差。这样，则有

$$\sum_{i=1}^{n}\lambda_i = \sum_{j=1}^{d}\lambda_i^{+} + \sum_{j=d+1}^{n}\lambda_i^{0}$$

用式 $\operatorname{tr}(X^tX) = \sum_{i=1}^{n}\lambda_i$ 减去式 $\operatorname{tr}(X^{+t}X^{+}) = \sum_{i=1}^{d}\lambda_i$，可得

$$\sum_{i=1}^{m}\sum_{j=d+1}^{n}(x_{ij}^{2} - x_{ij}^{+2}) = \sum_{j=d+1}^{n}\lambda_i^{0} = \sum_{i=1}^{m}\sum_{j=d+1}^{n}\sigma_{ij}^{0\,2} \tag{6-229}$$

式(6-229)说明，只要主因子数 d 能正确确定，则后$(n-d)$个所剩的特征值 $\lambda_i^{0}(j=1,2,\cdots,d)$之和实际上是完全由误差的平方和所构成。

仿此我们可定义由式(6-221)所表示的纯误差在整个因子空间$\{q_j(j=1,2,\cdots,n)\}$上的特征值之和：

$$\sum_{j=1}^{n}\lambda_i^{e} = \sum_{j=1}^{d}\lambda_i^{e+} + \sum_{j=d+1}^{n}\lambda_i^{e0} \tag{6-230}$$

在此，以上标 e 来表示是来自误差。所以

$$\sum_{j=1}^{d} \lambda_j^{e+} = \sum_{i=1}^{m} \sum_{j=1}^{d} (\sigma_{ij}^{+})^2 \tag{6-231}$$

$$\sum_{j=1}^{d} \lambda_j^{e0} = \sum_{i=1}^{m} \sum_{j=d+1}^{n} (\sigma_{ij}^{0})^2 \tag{6-232}$$

这是因为在进行正交变换时，误差的协方差阵的迹也是不变的。结合式(6-230)到式(6-232)，又因 $\sum_{i=1}^{m}\sum_{j=1}^{n} (e_{ij})^2 = \sum_{j=1}^{d} \lambda_j^{e}$

于是有

$$\sum_{i=1}^{m}\sum_{j=1}^{n} (e_{ij})^2 = \sum_{i=1}^{m}\sum_{j=1}^{d} (\sigma_{ij}^{+})^2 + \sum_{i=1}^{m}\sum_{j=d+1}^{n} (\sigma_{ij}^{0})^2 \tag{6-233}$$

式(6-233)左边为量测误差平方和，它同样可表示为误差在 n 个全因子空间的投影平方和，其中第一项代表量测误差在主因子轴上的投影平方和，它在因子分析中不能被除去；而第二项则代表量测误差在次因子轴上的投影平方和，它在因子分析过程中可以被剔除。这三项都与所谓的剩余标准偏差有关，即

$$mn(RSD)^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} (e_{ij})^2 \tag{6-234a}$$

$$md(RSD)^2 = \sum_{i=1}^{m}\sum_{j=1}^{d} (\sigma_{ij}^{+})^2 \tag{6-234b}$$

$$m(n-d)(RSD)^2 = \sum_{i=1}^{m}\sum_{j=d+1}^{n} (\sigma_{ij}^{0})^2 = \sum_{j=d+1}^{n} \lambda_j^{0} \tag{6-234c}$$

式（6-234）的 3 个表达式告诉我们，可以从不同的途径来求 RSD。将式（6-234）三式代入式（6-233），并两边同除以 nm 得，

$$(RSD)^2 = (d/n)(RSD)^2 + [(n-d)/n](RSD)^2 \tag{6-235}$$

这一重要的恒等式就总结了上述的误差理论。在因子分析中，RSD 可被分为两项：置入误差（IE）和抽出误差（XE）。换言之，剩余标准误差（RSD），即真实误差（RE），可由一勾股定理的方式表出：

$$(RE)^2 = (IE)^2 + (XE)^2$$

式中

$$RE = RSD$$

$$IE = \sqrt{d/n}(RSD)$$

$$XE = \sqrt{[(n-d)/n]}(RSD)$$

式(6-234c)十分重要，是它将 RSD(或 RE)与可通过计算机运算求得的特征值 $\lambda_j(j=1,2,\cdots,n)$联系起来。这说明只需算出了协方差阵的特征值，就可将 RE、IE、XE 都分别求出来，为确定体系的独立组分数 d 奠定了基础。

（三）主因子数确定方法

由于存在量测误差，要精确确定未知混合体系中的独立组分数（主因子数）的确是一相当困难的任务。为解决这一问题，已提出很多种方法，它们可分为两大类：① 基于量测误差大小已知的方法；② 对量测误差大小无可估计，只根据计算所得结果来作出近似判断的方法。下面分别对它们作出介绍。

1. 基于量测误差大小已知的方法

这些方法的基点是假设量测误差已知，然后与算出的量测数据矩阵 **X** 的特征值，并根据某一原则取某个主因子数所求出的误差大小进行比较，如该算出的真实误差（RE）落入了已知量测误差的范围，则据此可确定主因子即体系的独立组分数。目前基于量测误差大小已知的方法中应用较多的是

剩余标准偏差法（真实误差法，RE 法）。

该法由 Malinowski 提出，其计算式由式（6-236）给出，即

$$m(n-d)(RSD)^2=\sum_{i=1}^{m}\sum_{j=d+1}^{n}(\sigma_{ij}^{*})^2-\sum_{j=d+1}^{n}\lambda_j^0 \qquad (6\text{-}236)$$

式中，RSD 与真实误差（RE）是同一的，从式（6-234a）可以看出，它们的大小就代表了e_{ij}均方差的大小，将式（6-236）重新排列可得

$$RE^2=RSD^2=\sum_{j=d+1}^{n}\lambda_j^0/[m(n-d)] \qquad (6\text{-}237)$$

上式提供了一个很好的确定主因子数的标准，其判别过程如下：

首先先把最大特征值所对应的特征向量看成主因子轴（主轴），其余的都看成次要因子（次轴）而包括在上式中来计算 RSD，并将此计算值与已知的量测误差进行比较。如果RSD 大于已知的量测误差，那么则说明主因子数目就等于 1，该因子空间是一维的。仿此可继续考虑第二大的特征值，如此时 RSD 仍然大于已知量测误差，则需继续考虑第三个特征值所对应的特征向量，如此下去直至当取到某个特征值所对应的特征向量，它所对应的 RSD 已与所知量测误差近似相等，这时所已包含的主轴中的特征向量的个数即为体系中所含独立组分的个数。

2. 量测误差大小未知的方法

这些方法毋需已知量测误差，但因一般可设量测误差为零均等方差且服从正态分布的不相关白噪声，故可据此设计一些经验函数来估计主因子数。Malinowski 设计的因子指示函数 IND 法[141]和 Fisher 方差比[142]就是这样的例子，可用于在量测误差未知时确定主因子即体系中的独立组分数。

（1）因子指示函数法（IND 法）　因量测误差未知，不可能像上述方法那样采用比大小方法来确定主因子轴，所以 IND 法是利用某一原则来构造一个因子数的函数，随着主因子数的变化这个函数将在某一点上达到极小，此时，该极小点所对应的就是主因子数。为此他定义了如下的 IND 函数：

$$\mathrm{IND}=RSD/(n-k)^2 \qquad (6\text{-}238)$$

利用这一 IND 指示函数，可从 n 个特征值 λ_j（$j=1, 2, \cdots, n$）中，逐一计算不同的 k 所对应的 IND 值，即将式（6-237）定义的 RSD 代入式（6-238）。当 k 为真实的主因子数时，IND 函数将出现最小值，此最小值所对应的 k 即为我们要求的主因子数（混合体系的独立组分数）。值得指出的是，对于一般纯组分光谱重叠不严重的体系，且量测误差确为零均等方差且不相关白噪声时，此法确可得到正确的主因子数，但是当体系纯组分光谱重叠严重时，或量测误差为不等方差的噪声时，此法难于得到正确答案。表 6-13 示出了一个用 IND 函数来判别主因子数的实例。

（2）Fisher方差比法　由前述因子分析的误差理论可知，特征值 λ_j（$j=1, 2, \cdots, n$）就代表量测矩阵的方差，所以，它们的比值就可相当于 Fisher 方差比；另一方面，从统计意义上讲，主轴（primary axes）和次轴（secondary axes）所对应的特征值代表了不同的意义，即分别代表了组分光谱信息和量测误差的信息，故这两种特征值应存在显著性差异，Fisher 方差

比法的基本原理就在于此。然而，Fisher 方差比计算要求被判断的样本相互独立且服从正态分布，因所有的因子轴（即特征向量）是相互正交的，故独立性这一条件完全满足，而数据的残差一般都服从正态分布，所以，特征值比检验对上述的两个统计条件是基本上满足的。根据式（6-228），可将误差特征值（次轴）对应方差表述为

$$\mathrm{Var}(0)=\sum_{j=k+1}^{n}\lambda_j/(n-k) \tag{6-239}$$

表 6-13 体系中组分数的 IND 法确定

j	λ_j	RSD	IND×10^5	j	λ_j	RSD	IND×10^5
1	19.193396	0.015461	24.16	6	0.000260	0.000787	8.74
2	0.368079	0.003208	6.55	7	0.000141	0.000760	19.0
3	0.009065	0.002110	5.86	8	0.000132	0.000704	70.4
4	0.004414	0.000960	3.85	9	0.000099		
5	0.000294	0.000889	5.56				

对应于主轴的特征值主要来自组分量测谱的贡献，因显著地大于误差特征值（次轴）对应的方差，可记为 Var(k)。而特征值一般都是按大小顺序排列，因此可以这样来构造 Fisher 检验：

$$F(1,n-k)=\mathrm{Var}(k)/\mathrm{Var}(0)=\lambda_k(n-k)/\sum_{j=k+1}^{n}\lambda_j \tag{6-240}$$

式中，1 和（$n-k$）分别表示 Fisher 方差比检验的自由度，这是因为每一个特征值对应于一个自由度。将 k 从 n 取起，对应不同的 k 值，都进行一次 Fisher 方差比检验，直至所取的 k 得到了具有显著性差异结果为止。

值得指出的是，以上的主因子数的确定方法都是从用整个矩阵的特征值（它们反映了整个协方差阵的方差）来考虑的，对于这样的方法，常被称为全局因子分析法。在以后的讨论中可以看到，对于一些组分量测光谱重叠不严重且组分相对浓度相差不大的混合物体系，这类方法一般可以得到真实解。但是，对于一些组分量测光谱重叠严重且组分相对浓度相差很大的混合物体系，此类方法就一般不可靠了，此时宜用局部因子分析法来估计体系的独立主因子数。有关这一结论的论述，将在以后的讨论中逐步展开。

另外，因为特征向量，其中特别是载荷矢量就反映了组分量测谱的综合信息，所以，主因子数的确定方法也可以从特征向量的角度来考虑，此法的要点就是直接观察特征矢量，那些只包含误差信息的特征向量应该就对应于次轴。有关这一点的论述，也将在以后的讨论中逐步展开，并给出实例加以说明。

二、矩阵分辨方法

在确定了体系的独立组分数以后，对黑色分析体系的解析的下一步就是分辨出纯物种的光谱了。最早提出的光谱分辨基本思想是自模式曲线分辨法（SMCR）[143]。根据 Lambert-Beer 定律，本节所讨论的混合物量测谱矩阵 $\boldsymbol{X}$，实际上可表为两个矩阵的乘积，即

$$\boldsymbol{X}=\boldsymbol{C}\boldsymbol{S}^{t}+\boldsymbol{E} \tag{6-241}$$

其中，$\boldsymbol{C}$ 称为浓度分布矩阵，它是由每个不同纯组分的浓度矢量构成，为一（$m\times d$）阶的矩阵；而 $\boldsymbol{S}$ 则为纯组分的量测光谱所组成，为一（$n\times d$）阶矩阵。对黑色分析体系进行分辨的主要目的就在于求得浓度分布矩阵和纯组分的量测光谱矩阵。自模式曲线分辨法的主要思路

就是先求出纯物种的光谱，亦即纯组分的量测光谱矩阵 S，也就是先将黑色分析体系“白化”，继利用最小二乘法即可求得浓度分布矩阵 C，以完成对黑色分析体系的同时定性定量解析。

因在本书将要讨论的黑色分析体系的分辨方法大都源出于自模式曲线分辨法，所以首先将对此法进行详细的介绍。另外，基于自模式曲线分辨法而发展的很多方法也主要是基于矩阵类型的数据，即（6-241）式，所以，将它们都统称为矩阵分辨方法，以便与基于张量数据的分辨方法进行区别。

（一）自模式曲线分辨法

自模式曲线分辨法主要基于主成分分析（PCA），其基本假设只有两条：① 混合物量测谱符合线性加合性；② 波谱量测值只能为正值。因这两条假设要求条件对一般波谱仪器所产生的数据皆可成立，故其适用性较广，该法最早由 Lawton 和 Sylvestre 提出[143]，他们以两组分混合物体系的纯物种光谱解析为其目标，以得到它们的可行解、后被拓广至三组分或三组分以上的混合体系。因随着组分数目的增加，纯组分光谱的可行解域确定的困难显著增加，为叙述方便，本书将其分成两步加以介绍。

1．两组分体系的纯组分波谱辨析

【基本思路】

对于任一两组分混合物波谱，根据 Lambert-Beer 定律，可表示为

$$\boldsymbol{x}_i^{t}=c_{i1}\boldsymbol{s}_1^{t}+c_{i2}\boldsymbol{s}_2^{t}\qquad(i=1,2,\cdots,m)\tag{6-242}$$

式中，$s_j^t(j=1,2)$为未知组分的纯物种波谱矢量，由在某特定波长区域所测得的波谱非负量测值组成，且满足 $\sum\limits_{i=1}^{n}s_{ij}=1$，在此，$n$ 为波谱量测点数；c_{i1} 和 c_{i2} 分别表示两纯物种的相对浓度，且满足 $c_{i1}\neq k\,c_{i2}$，k 为任意比例常数，此条件说明各混合物中的相对浓度应不尽对应成比例；m 为混合样本个数。一般要求 m 和 n 都要大于体系的纯组分数（此例为 2）。

总的说来，自模式曲线分辨法的第一步是对数据矩阵 X 进行主成分分析，然后，通过前述的两个约束条件来求出 $s_j^t(j=1,2)$的可行解域。

【基本算法】

两组分体系自模式曲线分辨法的具体计算步骤如下：

(1) 计算混合样本组成的量测矩阵的协方差阵，继对其进行正交变换，分别求出其特征值和特征向量，

$$\boldsymbol{Q}^{t}\boldsymbol{X}^{t}\boldsymbol{X}\boldsymbol{Q}=\begin{bmatrix}\lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0\\ \vdots & \vdots & & \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & \lambda_d & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0 & \lambda_{d+1} & \cdots & 0\\ \vdots & \vdots & & \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & 0 & 0 & \cdots & \lambda_n\end{bmatrix}\tag{6-243}$$

式中，Q 为列向量相互正交的特征向量矩阵，λ_i 为相应的特征值，并设 $\lambda_i\geqslant\lambda_{i+1}$ $(i=1,2,\cdots,n-1)$。

从上一节讨论的结果可知，对于二组分混合体系，我们知道矩阵 X 的秩为 2，即除 λ_1 和 λ_2 外，其余特征值都会接近于零。取 λ_1、λ_2 所对应的两个特征向量 $\boldsymbol{q}_1$、$\boldsymbol{q}_2$ 组成新的特征向量矩阵 Q^*，即

$$Q^{*}=[q_1,q_2] \tag{6-244}$$

正交矩阵 Q^{*} 是一个 $(n\times2)$ 阶的矩阵，它的每一列都含有 n 个元素，即相当于波长的量测点数目，在因子分析中常被称为载荷矢量（loadings），在多元分辨中也常称为抽象量测谱。

从线性代数的知识可知（参见第十一章），由 s_1、s_2 所张成的线性空间 $\psi(s)$ 与由 q_1,q_2 所张成的线性空间 $\psi(q)$ 相等，即 $\psi(s)=\psi(q)$，它们可以相互线性表出，

$$s_j^{t}=t_{j1}^{t}q_1+t_{j2}^{t}q_2 \quad (j=1,2) \tag{6-245}$$

再从（6-242）式可知

$$\begin{aligned}x_i^{t}&=c_{i1}s_1^{t}+c_{i2}s_2^{t}\\&=c_{i1}(t_{11}q_1^{t}+t_{12}q_2^{t})+c_{i2}(t_{21}q_1^{t}+t_{22}q_2^{t})\\&=(c_{i1}t_{11}+c_{i2}t_{21})q_1^{t}+(c_{i1}t_{12}+c_{i2}t_{22})q_2^{t}\\&=u_{i1}q_1^{t}+u_{i2}q_2^{t} \qquad (i=1,2,\cdots,m)\end{aligned}$$

因 q_1，q_2 及 x_i^{t} 都为已知，且 q_1，q_2 相互正交，故可容易求出 u_{i1}、u_{i2}来，

$$\begin{aligned}x_i^{t}q_1&=u_{i1}q_1^{t}q_1+u_{i2}q_2^{t}q_1\\&=u_{i1}q_1^{t}q_1\end{aligned}$$

则有

$$u_{i1}=x_i^{t}q_1/q_1^{t}q_1=x_i^{t}q_1 \tag{6-246}$$

同理可得

$$u_{i2}=x_i^{t}q_2/q_2^{t}q_2=x_i^{t}q_2 \tag{6-247}$$

在此利用了正交矩阵的性质：$q_1^{t}q_2=0$ 和 $q_i^{t}q_i=1(i=1,2)$。

（2）约束条件的计算

① 波谱量测值只能为正值之约束条件的计算：

从式（6-245）可知

$$s_{ji}=t_{j1}q_{1i}+t_{j2}q_{2i}\geqslant0 \quad (j=1,\ 2;\ i=1,\ 2,\ \cdots,\ n) \tag{6-248}$$

亦即

$$t_{j2}\geqslant-\min|q_{1i}/q_{2i}|t_{j1}\text{当} \qquad q_{2i}>0 \tag{6-249a}$$

$$t_{j2}\leqslant\max|q_{1i}/q_{2i}|t_{j1}\text{当} \qquad q_{2i}<0 \tag{6-249b}$$

上述不等式（6-249）之所以成立，是因为用主成分分析方法所得的第一特征矢量 q_1 的每一元素 q_{1i}（$i=1$，2，…，n）皆大于或等于零。

② 组分浓度值只能为正值之约束条件的计算：

仍从式（6-245）出发，可有

$$s_1=t_{11}q_1+t_{12}q_2 \tag{6-250a}$$

$$s_2=t_{21}q_1+t_{22}q_2 \tag{6-250b}$$

如采用矩阵的表示形式，式（6-250）可变为

$$\begin{bmatrix}s_1\\s_2\end{bmatrix}=\begin{bmatrix}t_{11}&t_{12}\\t_{21}&t_{22}\end{bmatrix}\begin{bmatrix}q_1\\q_2\end{bmatrix} \tag{6-251}$$

另外，根据 $x_i^{t}=c_{i1}^{t}s_1+c_{i2}^{t}s_2 \quad (i=1,\ 2,\ \cdots,\ m)$，有

$$x_i=[c_{i1}\quad c_{i2}]\begin{bmatrix}s_1\\s_2\end{bmatrix}=[u_{i1}u_{i2}]\begin{bmatrix}q_1\\q_2\end{bmatrix} \tag{6-252}$$

将式(6-251)代入式(6-252)可得

$$\boldsymbol{x}_i=[c_{i1}c_{i2}]\begin{bmatrix}t_{11} & t_{12}\\ t_{21} & t_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{q}_1\\ \boldsymbol{q}_2\end{bmatrix}=[u_{i1}u_{i2}]\begin{bmatrix}\boldsymbol{q}_1\\ \boldsymbol{q}_2\end{bmatrix} \tag{6-253}$$

所以有

$$[c_{i1}c_{i2}]\begin{bmatrix}t_{11} & t_{12}\\ t_{21} & t_{22}\end{bmatrix}=[u_{i1}u_{i2}] \tag{6-254}$$

即

$$\begin{bmatrix}t_{11} & t_{12}\\ t_{21} & t_{22}\end{bmatrix}=\begin{bmatrix}c_{i1}\\ c_{i2}\end{bmatrix}=\begin{bmatrix}u_{i1}\\ u_{i2}\end{bmatrix}$$

$$\begin{bmatrix}c_{i1}\\ c_{i2}\end{bmatrix}=\begin{bmatrix}t_{11} & t_{12}\\ t_{21} & t_{22}\end{bmatrix}^{-1}\begin{bmatrix}u_{i1}\\ u_{i2}\end{bmatrix}=\frac{1}{t_{11}t_{22}-t_{12}t_{21}}\begin{bmatrix}t_{22} & -t_{21}\\ -t_{12} & t_{11}\end{bmatrix}\begin{bmatrix}u_{i1}\\ u_{i2}\end{bmatrix}$$

对于 $t_{12}t_2>t_{11}t_{22}$ 的情况，有

$$\begin{bmatrix}t_{22} & -t_{21}\\ -t_{12} & t_{11}\end{bmatrix}\begin{bmatrix}u_{i1}\\ u_{i2}\end{bmatrix}\leqslant 0$$

写成标量形式则有

$$t_{12}\geqslant -\max_{1\leqslant i\leqslant n}|u_{i2}/u_{i1}|t_{11} \tag{6-255a}$$

$$t_{22}\leqslant \min_{1\leqslant i\leqslant n}|u_{i2}/u_{i1}|t_{21} \tag{6-255b}$$

（3）可行解域的图像表示　从上述讨论可以看出，纯物种波谱 $s_j^t(j=1,2)$ 及混合物波谱矢量 $x_i^t(i=1,2,\cdots,n)$ 皆可由它们在 q_1,q_2 组成的二维空间坐标表出，而不等式(6-251)及式(6-255)又可由 q_1,q_2 组成的二维空间平面中的直线表出。借计算机图像显示功能，即可在二维平面上将纯物种波谱 $s_j^t(j=1,2)$ 的可行解域表示出来，只要确定了 $s_j^t(j=1,2)$ 在此二维平面上的坐标（t_{j1}，t_{j2}），即可根据式（6-245）求出 s_j^t 的可能波谱形态。下以 Lawton 给出的一个实例对上述计算过程加以具体说明。

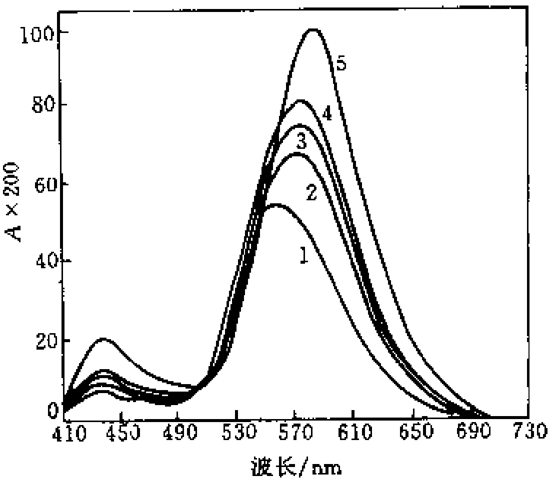


图 6-21　5个两组分混合物的可见吸收光谱图

由图 6-21 所示的 5 条可见吸收光谱曲线是对两种染料的混合样本量测所得，它们的具体吸收光谱矢量是由从 410nm～700nm 每隔 10nm 读取量测数据面得，组成混合样本量测矩阵 $\boldsymbol{X}$，继对它的协方差阵进行正交分解，取两个对应于最大特征值的特征向量，由式(6-246)及式(6-247)可分别算得混合物波谱矢量 x_i^t 在 q_1、q_2 组成的二维空间上的坐标，它们分别为 x_1^t(125.7，−33.4)、x_2^t(201.7，−6.6)、x_3^t(186.5，−13.2)、x_4^t(169.3，−13.2)、x_5^t(235.4，39.5)。按此计算的结果可得由 q_1,q_2 组成的二维平面图示于图 6-22，用（6-251）式可得两个组分的可行解域的外约束界 F_1 和 F_2；用（6-255）式可得两个组分的可行解域的外约束界 F_1^* 和 F_2^*。

因 $\sum\limits_{i=1}^{n}\sum s_{ij}^t=1$，利用这一条件，由式（6-245）可得

$$\sum_{i=1}^{n} s_{ij}^{t}=t_{j1}\sum_{i=1}^{n} q_{1i}+t_{j2}\sum_{i=1}^{n} q_{2i}=1 \quad (j=1, 2)$$

令 $\sum q_{1i}=a$，$\sum q_{2i}=b$ 则有

$$t_{j1}a+t_{j2}b=1$$

这说明，正规化为 1 的 s_j^t（$j=1$，2）必然落在由 $\boldsymbol{q}_1$，$\boldsymbol{q}_2$ 组成的二维空间上的一条直线上，它就是图 6-22 中的 $A_{\rm I}$ 和 $A_{\rm II}$ 所表示的线段。因 $A_{\rm I}$ 和 $A_{\rm II}$ 分别落在可行解域中，所以，s_j^t（$j=1$，2）也就必然落在 $A_{\rm I}$ 和 $A_{\rm II}$ 表示的可行解域中。求得 s_j^t（$j=1$，2）在 $\boldsymbol{q}_1$，$\boldsymbol{q}_2$ 组成的二维空间中的坐标，就可容易地算得它们的可行解域来。图 6-23 所示的 $F_{\rm I}$ 和 $F_{\rm II}$ 就将它们具体表示出来。

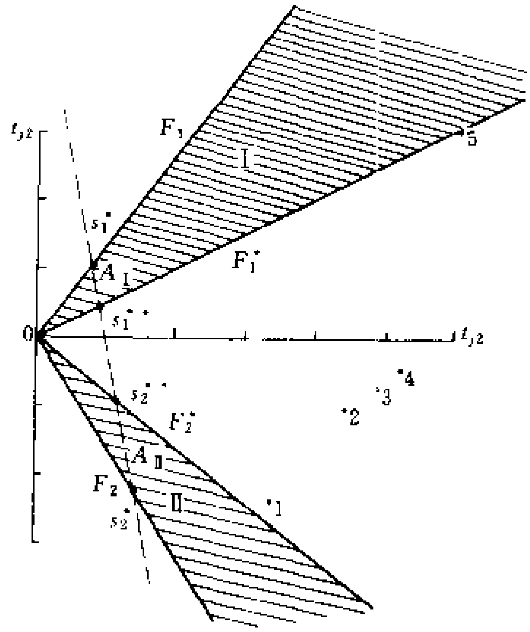


图 6-22　由抽象量测谱 q_1-q_2 组成的平面投影图
阴影部分 I 和 II 分别表示两个组分的可行解域，其中直线 $A_{\rm I}$ 和 $A_{\rm II}$ 分别为两个组分光谱 s_1 和 s_2 的可行解域

【讨论】

这一方法提出后，由 Kowalski 等将其用于气相色谱-质谱[144]及高效液相色谱-紫外可见光谱数据的解析[145]，分别分析了 α-蒎烯和蒎烷及烟酸胺和核黄素的重叠色谱峰，获满意结果。说明用此法可分辨用色谱法也难以很好分离的两组分体系。

Kawata 等[146]又分别研究了实际量测数据所含误差对可行边界确定的影响以及在可行解域中确定最可能纯组分光谱的问题，提出用最小熵的方法找到纯组分光谱的最可能解。Koenig[147]等对在分辨时常可碰到的一种特殊情况，即某一组分在一段波长范围内有其自身特殊的吸收峰的两组分分辨情况，这在红外光谱指纹区很是多见，提出一种直接确定纯组分唯一解的方法。本节还将对这些内容分别加以介绍。

(1) 噪声对可行解域边界确定的影响及其克服方法　在前文的讨论中，我们没有考虑噪声的影响，而在实际上量测误差的存在必将影响到抽象量测谱中的元素的数值，这种误差即为在前节因子分析误差理论中曾讨论过的置入误差。从式（6-31）可以看出，它将影响到可行解域边界的确定，有时甚至可将真实解排除出可行解域之内。为此，Kawata 将滤波的方法引入，以排除 $q_{i1}(i=1,2,\cdots,n)$ 中小于噪声的那些数值，即对 q_{i1}' 用（6-256）式进行重新计算：

$$q_{i1}'=(q_{i1}^2+\sigma_n^2)/q_{i1} \quad (i=1,2,\cdots,n) \tag{6-256}$$

式中，σ_n 为一常数。它可由式（6-257）给出

$$\sigma_n=\sqrt{(1/n)\sum_{i=1}^{n}(q_{i1})^2/(p_s/p_n)} \tag{6-257}$$

式中，(p_s/p_n) 的定义为

$$(p_s/p_n)=\lambda_1/\left[\sum_{k=3}^{n}\lambda_k/(n-2)\right] \tag{6-258}$$

式中，$\lambda_k(k=1,2,...,n)$为主成分分析所求得的特征值，且有$\lambda_k \geqslant \lambda_{k+1}$。图 6-24 示出了 q'_{i1}与 q_{i1}之间的关系。从图 6-24 可以看出，如果 q_{i1}小于 σ_n 则被滤掉 σ_n 而代之以 q_{i1}，这样也就消除了噪声对可行解域边界确定的影响。

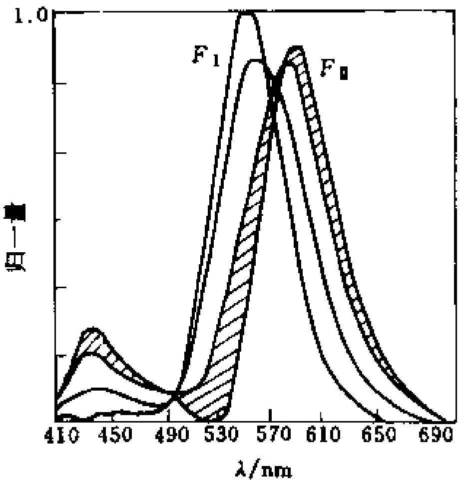


图 6-23　两个纯组分光谱 s_1 和 s_2 的可行解域

F_{I}：s_1的可行解域；F_{II}：s_2的可行解域

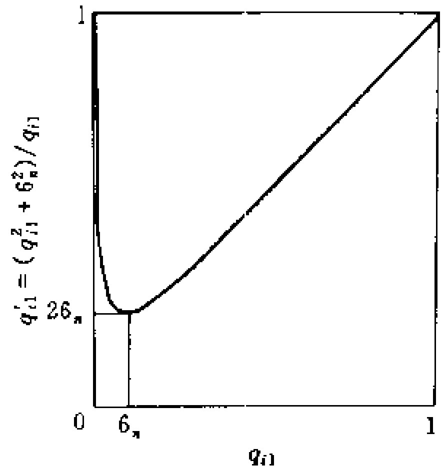


图 6-24　q'_{i1}与 q_{i1}的关系图

(2) 最小熵确定纯组分光谱　在前述讨论中知道，Lawton 和 Sylvestre 提出的自模式曲线分辨法只能求得各组分的可行解域。试想如果能依照某一原则在可行解域中找到一个合理的唯一解，这将对自模式曲线分辨法的实际运用具有重要意义。为此，Kawata 等引入了最小熵的概念，即用组分光谱自身所具有的特性来确定最可能的纯组分光谱唯一解。他们在实验中发现，在所有纯光谱的可行解域中，那些不是实际纯组分光谱的可行光谱实际是由纯组分光谱与另一纯组分光谱的线性组合而成，从而导致纯组分的导数光谱的混合程度（即熵）增加，故提出用导数光谱的熵来确定唯一解。其具体求解过程如下：

$$H_j = -\sum_{i=1}^{n} p_{ij}\lg(p_{ij}) \rightarrow \text{极小} \qquad (j=1,2)$$

式中，p_{ij}为第 j 个被故纯组分光谱在波长点 i 处的二阶或一阶导数，并经式(6-259)正规化，

$$p_{ij} = |s'_{ij}| / \sum_{i=1}^{n} |s'_{ij}| \qquad (i=1,2,\cdots,n) \tag{6-259}$$

式中，s'_{ij}表示可行解域中第 j 个被故纯组分光谱 s_j 在波长点 i 处的二阶或一阶导数，经此正规化处理，p_{ij}就具有概率函数的性质了。

从图 6-22 可知，s_1 和 s_2 的可行解域 A_{I} 和 A_{II} 可由式（6-260）表示，即

$$s_1 = s_1^* + (1-\gamma_1)s_1^{**} \qquad 0 \leqslant \gamma_1 \leqslant 1 \tag{6-260a}$$

$$s_2 = s_2^* + (1-\gamma_1)s_2^{**} \qquad 0 \leqslant \gamma_1 \leqslant 1 \tag{6-260b}$$

借一维寻优的黄金分割法(亦称 0.618 法)，以 $H_j = -\sum p_{ij}\lg(p_{ij})$作为目标函数即可求出 s_1 和 s_2 的唯一解来。

Kawata 等用上述方法对甲基橙和溴甲酚绿的两组分体系进行解析，同时还对噪声影响可行解域及用最小熵求唯一解也作了探讨，所得结果分别示于图 6-25 至图 6-27。从图 6-26 可以看出，如果不考虑噪声的影响，的确可能将真正纯组分光谱排除在可行解域之外，图中溴甲酚绿纯组分光谱的可行解域就正是这种情况。在图 6-27 中，在考虑了噪声的影响，并用前述滤波方法对其进行滤波后，确实克服了噪声的影响。另外从图 6-27 中，还可以看到，用最小熵方法

所找到的纯组分光谱与图 6-25 中的实际光谱比较，的确吻合较好，说明用最小熵为目标函数寻优的方法具有一定实际意义。

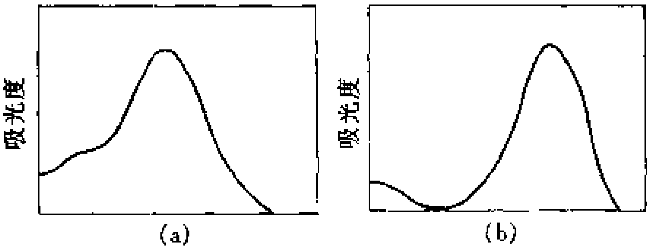


图 6-25 溴甲酚绿和甲基橙的标准光谱图

(a) 溴甲酚绿(*BG*)；(b) 甲基橙(*MO*)

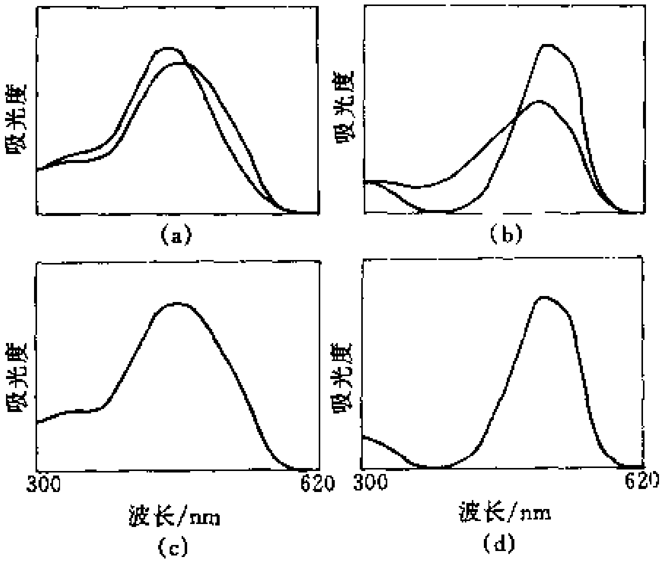


图 6-26 没进行噪声滤波的可行解域和计算所得各组分光谱图

(a) 溴甲酚绿的可行解域；(b) 甲基橙的可行解域；

(c) 用最小熵方法算出的溴甲酚绿的纯组分光谱；(d) 用最小熵方法算出的甲基橙的纯组分光谱

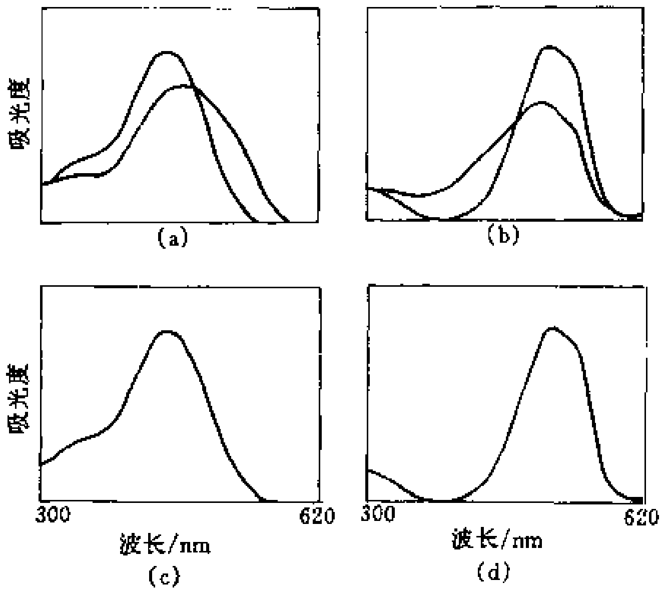


图 6-27 进行噪声滤波后的可行解域和计算所得各组分光谱图

(a) 溴甲酚绿的可行解域；(b) 甲基橙的可行解域；

(c) 用最小熵方法算出的溴甲酚绿的纯组分光谱；(d) 用最小熵方法算出的甲基橙的纯组分光谱

2. 三组分体系的纯组分波谱辨析

最早用自模式曲线分辨法来进行三组分体系的纯物种波谱解析的要首推 M. Ohta[148]，他以蓝青色、洋红和黄色这三种用于彩色照相中的染料作为分辨对象，用自模式曲线分辨法找到了这三种纯组分光谱的可行解域，并用 Monte-Carlo 方法找到了一组可能组分光谱。Sharaf[149]在此基础上对三组分体系的纯组分光谱的自模式分辨进行了进一步的研究，并对其可行解域的降维表示技术和定量分析进行了阐述。

与二组分体系的纯组分波谱辨析的思路类似，自模式曲线分辨法对三组分体系解析的第一步仍是对数据矩阵 X 进行主成分分析，然后，通过两个约束条件来求出 s_j^t ($j=1, 2, 3$) 的可行解域。

对混合样本组成的量测矩阵的协方差阵进行正交变换，分别求出其特征值和特征向量，三组分混合体系，我们知道矩阵 X 的秩为 3，就取 λ_1、λ_2 和 λ_3 所对应的三个特征向量 q_1、q_2 和 q_3 组成新的特征向量矩阵 Q^*，即

$$Q^*=[q_1,q_2,q_3] \tag{6-261}$$

在因子分析中 Q^* 被称为载荷矢量（loadings）矩阵，亦称抽象量测谱矩阵。

与前节内容相仿，有

$$s_j^t=t_{j1}q_1^t+t_{j2}q_2^t+t_{j3}q_3^t \qquad (j=1,2,3) \tag{6-262}$$

和

$$x_i^t=c_{i1}s_1^t+c_{i2}s_2^t+c_{i3}s_3^t \tag{6-263}$$

将式（6-262）代入（6-263）式，可得

$$\begin{aligned} x_i^t &= c_{i1}(t_{11}q_1^t+t_{12}q_2^t+t_{13}q_3^t)+c_{i2}(t_{21}q_1^t+t_{22}q_2^t+t_{23}q_3^t)+c_{i3}(t_{31}q_1^t+t_{32}q_2^t+t_{33}q_3^t) \\ &= (c_{i1}t_{11}+c_{i2}t_{21}+c_{i3}t_{31})q_1^t+(c_{i1}t_{12}+c_{i2}t_{22}+c_{i3}t_{32})q_2^t+(c_{i1}t_{13}+c_{i2}t_{23}+c_{i3}t_{33})q_3^t \\ &= u_{i1}q_1^t+u_{i2}q_2^t+u_{i3}q_3^t \qquad (i=1,2,\cdots,m) \end{aligned} \tag{6-264}$$

与前节对二组分体系的纯组分波谱自模式辨析的叙述不同，在这一节将对降维技术予以介绍。从式(6-264)可以看出，

$$u_{i1}=c_{i1}t_{11}+c_{i2}t_{21}+c_{i3}t_{31} \tag{6-265a}$$

$$u_{i2}=c_{i1}t_{12}+c_{i2}t_{22}+c_{i3}t_{32} \tag{6-265b}$$

$$u_{i3}=c_{i1}t_{13}+c_{i2}t_{23}+c_{i3}t_{33} \tag{6-265c}$$

如记 $u_i=[u_{i1},u_{i2},u_{i3}]^t, t_1=[t_{11},t_{12},t_{13}]^t, t_2=[t_{21},t_{22},t_{23}]^t, t_3=[t_{31},t_{32},t_{33}]^t$，则上述方程组（6-265）就可变为

$$u_i^t=c_{i1}t_1^t+c_{i2}t_2^t+c_{i3}t_3^t \tag{6-266}$$

将式(6-266)与式(6-263)比较，就可发现，两式有类似的形式。不同的是，式(6-263)为任一混合物量测光谱由 n 维纯组分光谱空间 $\{s_1,s_2,s_3\}$ 的线性组合之表出方式，而式(6-266)则为该混合物波谱在 n 维正交因子空间 $\{q_1,q_2,q_3\}$ 之坐标组成的向量 $u_i=[u_{i1},u_{i2},u_{i3}]^t$ 在一三维空间 $\{t_1,t_2,t_3\}$ 的线性组合之表出方式。然而，它们的线性组合系数是同一的，这说明任一混合物量测光谱在经主成分分析后，实际是可由一三维线性空间来表出的，而且这一关系在三维空间中还具有相同形式。以下还将证明，当满足一定条件时，混合物光谱在因子空间中完全落在一个平面上，这样，三组分混合体系的可行解域就可在一平面上表出，从而使得三组分混合体系的可行解域的确定将与前节所述的二组分混合体系的一样容易。

对于(6-34)式，当满足下述条件时，

$$c_{i1}+c_{i2}+c_{i3}=1 \qquad (i=1,2,\cdots,m) \tag{6-267}$$

则混合物光谱 x_i^t（$i=1$，2，…，m）在因子空间的坐标点集（u_{i1}，u_{i2}，u_{i3}），（$i=1$，2，…，m），必落在由三个纯组分光谱在因子空间的坐标点［参见式（6-33)］，即（t_{11}，t_{12}，t_{13}）、（t_{21}，t_{22}，t_{23}），（t_{31}，t_{32}，t_{33}）所构成的平面上。

由解析几何知识可知，三点决定一个平面，设由点（t_{11}，t_{12}，t_{13}），（t_{21}，t_{22}，t_{23}），（t_{31}，t_{32}，t_{33}）构成的平面方程如下：

$$\alpha x+\beta y+\gamma z=d \tag{6-268}$$

显然有

$$\alpha t_{11}+\beta t_{12}+\gamma t_{13}=d \tag{6-269a}$$

$$\alpha t_{21}+\beta t_{22}+\gamma t_{23}=d \tag{6-269b}$$

$$\alpha t_{31}+\beta t_{32}+\gamma t_{33}=d \tag{6-269c}$$

所谓混合物光谱 $x_i^t(i=1,2,\cdots,m)$在因子空间的坐标点集$(u_{i1},u_{i2},u_{i3}),(i=1,2,\cdots,m)$，必落在由$(t_{11},t_{12},t_{13}),(t_{21},t_{22},t_{23}),(t_{31},t_{32},t_{33})$所构成的平面之上，实质上就是要满足：

$$\alpha u_{i1}+\beta u_{i2}+\gamma u_{i3}=d \tag{6-270}$$

根据（6-264）式，将（u_{i1}，u_{i2}，u_{i3}）所对应的值代入式（6-270）的左边，可得

$$\begin{aligned}&\alpha(c_{i1}t_{11}+c_{i2}t_{21}+c_{i3}t_{31})+\beta(c_{i1}t_{12}+c_{i2}t_{22}+c_{i3}t_{32})+\gamma(c_{i1}t_{13}+c_{i2}t_{23}+c_{i3}t_{33})\\&=c_{i1}(\alpha t_{11}+\beta t_{12}+\gamma t_{13})+c_{i2}(\alpha t_{21}+\beta t_{22}+\gamma t_{23})+c_{i3}(\alpha t_{31}+\beta t_{32}+\gamma t_{33})\\&=c_{i1}d+c_{i2}d+c_{i3}d\\&=(c_{i1}+c_{i2}+c_{i3})d\\&=d\end{aligned}$$

在上述推导中利用了式（6-269）以及式（6-267）的条件。到此，唯一的问题就是由式（6-267）表出的条件在实际中怎样来实现。其实，对于由式（6-267）表出的条件是容易在实际中做到的，即只需对混合物量测谱 x_i^t（$i=1$，2，…，m）和纯组分量测谱 s_i^t（$i=1$，2，3）进行归一化处理就立即可以得到。即令

$$\sum_{j=1}^{n} x_{ij}=1,\ \sum_{j=1}^{n}\sum s_{ij}=1 \qquad (i=1,\ 2,\ \cdots,\ m;\ j=1,\ 2,\ \cdots,\ n)$$

由式（6-263），对混合物量测谱 x_i^t 的所有元素求和可得

$$\sum_{j=1}^{n} x_{ij}^t=c_{i1}\sum_{j=1}^{n} s_{1j}+c_{i2}\sum_{j=1}^{n} s_{2j}+c_{i3}\sum_{j=1}^{n} s_{3j} \tag{6-271}$$

根据$\sum x_{ij}=1$，$\sum s_{ij}=1$ 的条件，立即可得

$$1=c_{i1}+c_{i2}+c_{i3} \tag{6-272}$$

这就是由式（6-267）表出的条件。

值得提出的是，由于纯组分光谱 $s_i^t(i=1,2,3)$是待求的，所以这一平面方程实际是通过由混合物光谱 $x_i^t(i=1,2,\cdots,m)$在因子空间的坐标点集来得到的。从上述讨论可知，经归一化处理后的混合物光谱 $x_i^t(i=1$，2，…，m）在因子空间的坐标点集$(u_{i1},u_{i2},u_{i3}),(i=1,2,\cdots,m)$皆落在一个平面上，故可任取其中三点，如$(u_{i1},u_{i2},u_{i3}),(u_{j1},u_{j2},u_{j3}),(u_{k1},u_{k2},u_{k3})$来构成这一平面，在此下标 i、j、k 是属于下标集（$i=1$，2，…，m）中的任一不相同的三个下标，继用下式可得由式（6-269）所定义的平面方程，即

$$\begin{vmatrix} u_{i1}-x & u_{i2}-y & u_{i3}-z \\ u_{i1}-u_{j1} & u_{i2}-u_{j2} & u_{i3}-u_{j3} \\ u_{i1}-u_{k1} & u_{i2}-u_{k2} & u_{i3}-u_{k3} \end{vmatrix}=0 \qquad (6\text{-}273)$$

Sharaf 等就是用此法对两个三组分混合体系进行了解析。其中一个是正己烷、环己烷和戊烯醇构成的三组分混合体系的纯组分质谱解析，另一个是 Cr（Ⅲ）、Co（Ⅱ）、Ni（Ⅱ）构成的三组分混合体系的纯组分紫外可见光谱解析。这两个体系的各组分浓度及它们的混合物量测波谱在三维因子空间的坐标和在二维平面的坐标分别示于表 6-14 和表 6-15，它们在二维平面的投影图分别示于图 6-28 和图 6-29。

表 6-14　10 个模拟质谱混合体系的组成、因子得分（a_{i1}, a_{i2}, a_{i3}）和计算所得的二维平面坐标及纯物种质谱

混合物编号	组分相对浓度/%			因子得分			二维平面坐标	
	Ⅰ	Ⅱ	Ⅲ				x	y
1	10	10	80	123.924	−49.112	−14.726	74.813	110.723
2	80	10	10	122.785	45.947	12.892	20.306	13.841
3	10	80	10	120.969	48.754	−44.676	107.938	13.841
4	50	50	0	121.585	61.131	1.805	62.594	0.000
5	0	50	50	122.398	−6.768	40.071	101.528	69.202
6	50	0	50	123.695	−8.774	22.477	38.934	69.203
7	40	0	60	123.858	−22.354	14.246	46.721	83.043
8	30	0	70	124.021	−35.934	6.015	54.507	96.884
9	30	10	60	123.598	−21.953	1.736	59.240	83.043
10	40	50	10	123.436	−8.373	9.967	51.453	69.203
				计算所得的纯物种质谱				
Ⅰ	100	0	0	122.882	59.126	63.633	0.000	0.000
Ⅱ	0	100	0	120.287	63.136	−61.464	125.188	0.000
Ⅲ	0	0	100	124.509	−76.674	−18.679	77.868	138.405

表 6-15　紫外光谱混合体系的组成、因子得分（a_{i1}, a_{i2}, a_{i3}）和计算所得的二维平面坐标及纯物种光谱

混合物编号	组分相对浓度/%			因子得分			二维平面坐标	
	Ⅰ	Ⅱ	Ⅲ				x	y
1	10	10	80	2.790	0.033	0.248	0.174	2.405
2	80	10	10	0.895	−0.397	−0.261	0.098	0.398
3	10	80	10	0.751	0.389	−0.294	0.898	0.392
4	50	50	0	0.530	−0.014	−0.350	0.542	0.096
5	50	0	50	2.025	−0.262	0.023	0.024	1.568
6	0	50	50	1.930	0.302	0.004	0.579	1.573
7	35	25	40	1.709	−0.043	−0.061	0.298	1.286
				计算所得的纯物种光谱				
Ⅰ	100	0	0	0.534	−0.564	−0.358	0.000	0.000
Ⅱ	0	100	0	0.331	0.593	−0.406	1.176	0.000
Ⅲ	0	0	100	23.776	−3.545	5.832	−7.198	23.143

表 6-14 和表 6-15 中的因子空间坐标（u_{i1}，u_{i2}，u_{i3}），（$i=1, 2, \cdots, m$）可由式（6-264）求出，这是因为因子空间 $\{q_1, q_2, q_3\}$ 是相互正交的，

$$x_i^t q_1 = u_{i1} q_1^t q_1 + u_{i2} q_2^t q_1 + u_{i3} q_3^t q_1 \qquad (i=1,2,\cdots,m) \qquad (6\text{-}274)$$

由正交性，有 $q_2^t q_1=0$，$q_3^t q_1=0$，以及 $q_2^t q_3$，

所以

$$u_{i1}=x_i^t q_1/q_1^t q_1 \quad (6\text{-}275a)$$

$$u_{i2}=x_i^t q_2/q_2^t q_2 \quad (6\text{-}275b)$$

$$u_{i3}=x_i^t q_3/q_3^t q_3 \quad (6\text{-}275c)$$

至于怎样由因子空间坐标(u_{i1},u_{i2},u_{i3})，$(i=1,2,\cdots,m)$ 化为二维平面坐标，则可根据方程（6-273）来进行，或是通过坐标旋转，使 xy 平面与由（6-273）所定义的平面重合即可得到。

从图 6-28 可以看出，由点1°、2°、3°决定的三角形是由质谱量测值非负这一约束条件而得到的，而由点 1、2、3 则是由浓度非负这一约束条件而得到的。图中的阴影部分就表示三个纯组分质谱的可行解域。图 6-29 可类似解释。有关多组分可行解域的确定在下一节还将进行更详尽的介绍，在此不多讨论了。

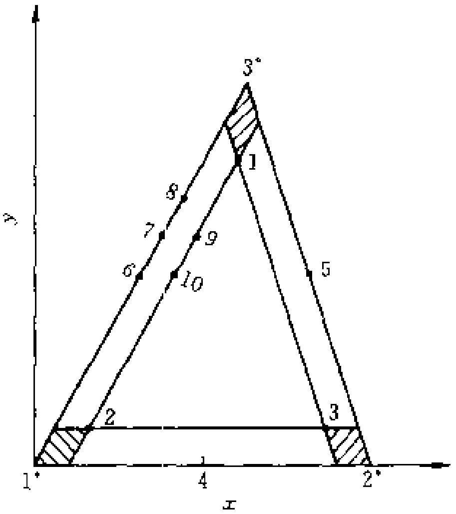


图 6-28　10个质谱的二维表示
点 1、2、3 表示内界；点 1°、2°、3°表示外界；
图中阴影区表示三个组分的可行解域

从以上讨论可以看出，自模式曲线分辨法的关键就是如何从一组混合物量测光谱中找到各个纯组分波谱的可行解域，而且三组分混合体系的解析与两组分混合体系的解析并无本质区别，只是在降维处理上更加复杂些而已。那么，对于自模式曲线分辨法是否可在降维技术上进行统一的处理，从而拓广至更多组分混合体系的纯组分波谱解析呢？这正是下一节将要介绍的内容。

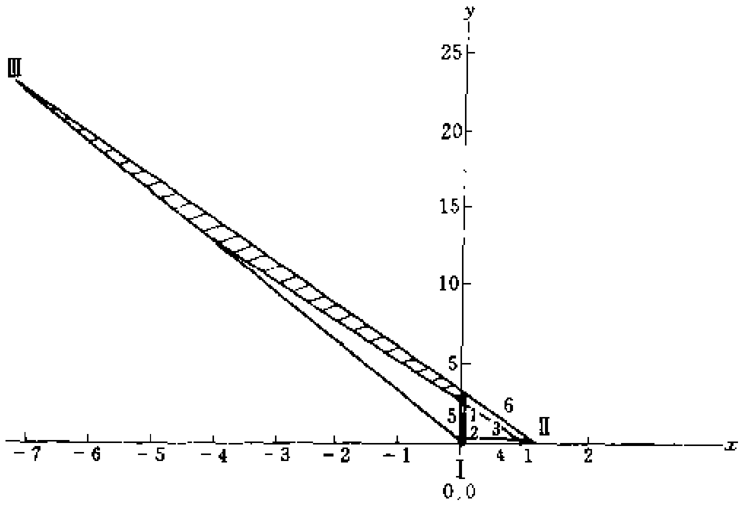


图 6-29　7 个紫外光谱的二维表示
点 1、2、3 表示下界；点 Ⅰ、Ⅱ、Ⅲ 表示上界；
图中的阴影部分表示三个组分光谱的可行解域

3. 自模式曲线分辨的统一降维处理和可行解域的确定

1985 年 Borgen 和 Kowalski 对此问题进行了研究[150]，对含 p 组分的混合物体系的分辨提出了统一的降维处理方法，说明含 p 组分的混合物体系纯组分波谱分辨的可行解域总可以在 $(p-1)$ 维超平面上表出。此外，该文还详细讨论了三组分体系之可行解域确定的一般性原则和方法。本节拟对这些内容分别加以介绍。

（1）降维处理变换

设对所得混合物量测波谱矩阵 X 进行了第一步的主成分分析，即通过对量测矩阵的协方

差阵正交分解后，找到了 p 个抽象波谱矢量 $\{q_1, q_2, \cdots, q_p\}$，并且还设已对所有的混合物量测波谱 x_i^t $(i=1, 2, \cdots, m)$ 和纯组分量测谱 s_j^t $(i=1, 2, \cdots, p)$ $(p<m)$ 进行了归一化处理，即

$$\sum_{j=1}^{n} x_{ij}=1, \qquad (i=1, 2, \cdots, m; j=1, 2, \cdots, n) \tag{6-276a}$$

$$\sum_{i=1}^{n} s_{ij}=1, \qquad (j=1, 2, \cdots, p; i=1, 2, \cdots, n) \tag{6-276b}$$

下面将看到，只需通过适当变换，就可将混合物量测波谱和纯组分量测谱中元素非负的约束条件在 $(p-1)$ 维的超平面上表示出。

首先来看看在此体系的任一向量(包括混合物量测矢量和我们要找的纯组分波谱矢量)怎样用 $(p-1)$ 维超平面表出的情况。因属于该体系的量测波谱矢量皆可由 p 个抽象波谱矢量线性表示出，即

$$s_j^t=t_{j1}q_1^t+t_{j2}q_2^t+\cdots+t_{jp}q_p^t(j=1,2,\cdots,p) \tag{6-277}$$

和

$$x_i^t=u_{i1}q_1^t+u_{i2}q_2^t+\cdots+u_{ip}q_p^t(i=1,2,\cdots,m) \tag{6-278}$$

再根据条件 $\sum x_{ij}=1$，对的各元素求和得

$$\sum_{j=1}^{n} x_{ij}^t=u_{i1}\sum_{j=1}^{n} q_{1j}+u_{i2}\sum_{j=1}^{n} q_{2j}+\cdots+u_{ip}\sum_{j=1}^{n} q_{pj} \tag{6-279}$$

令 $\sum_{j=1}^{n} q_{kj}=d_k$，则有

$$\sum_{k=1}^{p} u_{ik}d_k=1 \tag{6-280}$$

因第一正交抽象波谱矢量 q_1 的各元素总是非负，所以 d_1 非负，再将式 (6-280) 变形为

$$u_{i1}+(1/d_1)\sum_{k=2}^{p} u_{ik}d_k=1/d_1 \tag{6-281}$$

可得

$$u_{i1}=\left(1-\sum_{k=2}^{p}u_{ik}d_k\right)/d_1 \tag{6-282a}$$

对于由式 (6-277) 表示的 s_j^t $(j=1, 2, \cdots, p)$，上述推导亦同样成立，只需将 u_{ij}改成 t_{ij}即可，

$$t_{j1}=\left(1-\sum_{k=2}^{p}t_{jk}d_k\right)/d_1 \tag{6-282b}$$

式 (6-282) 说明，只要由归一化条件，就可将 u_{i1}及 t_{j1}由其余的 $u_{ik}(k=2,\cdots,p)$和 $t_{jk}(k=2,\cdots,p)$而求出，这就将 p 维空间降至了$(p-1)$维。

(2) 波谱数值非负约束 (第一约束，FIRPOL)

因所求纯组分波谱的强度值皆不可为负，此条件可用式 (6-283) 表示

$$s_{ij}\geqslant 0 \qquad (j=1,2,\cdots,p;i=1,2,\cdots,n) \tag{6-283}$$

又因为

$$s_j^t=t_{j1}q_1^t+t_{j2}q_2^t+\cdots+t_{jp}q_p^t \qquad (j=1,2,\cdots,p) \tag{6-284}$$

亦即

$$s_{ij}=t_{j1}q_{1i}+t_{j2}q_{2i}+\cdots+t_{jp}q_{pi}\geqslant 0 \tag{6-285}$$

再将式 (6-282b) 代入式 (6-285)，可得

$$s_{ij}=\Big[\Big(1-\sum_{k=2}^{n}t_{jk}d_k\Big)/d_1\Big]q_{1i}+t_{j2}q_{2i}+\cdots+t_{jp}q_{pi}$$
$$=q_{1i}/d_1-(q_{1i}/d_1)\sum_{k=2}^{n}t_{jk}d_k+\sum_{k=2}^{n}t_{jk}q_{ki}$$
$$=q_{1i}/d_1+\sum_{k=2}^{n}\sum\,[q_{ki}-(q_{1i}d_k)/d_1]t_{jk}\geqslant 0 \tag{6-286}$$

$q_{1i}/d_1\geqslant 0$，实因第一特征向量的每一元素都大于零，所以对式（6-286）两边同除以 q_{1i}/d_1，不等式不变号，得

$$1+\sum_{k=2}^{n}[(q_{ki}/q_{1i})d_1-d_k]t_{jk}\geqslant 0 \tag{6-287}$$

令$[(q_{ki}/q_{1i})d_1-d_k]=f_{ki}$，式（6-287）则变为

$$1+\sum_{k=2}^{n}f_{ki}t_{jk}\geqslant 0 \tag{6-288}$$

式（6-288）中 f_{ki}为一已知量，这是因为 q_{ki}（$k=2$，…，p，$i=1$，2，…，n）可由主成分分析而得到，而且 $d_k=\sum_{i=1}^{n}q_{ki}$，式中的未知变量就只有 t_{kj}（$k=2$，…，p，$j=2$，…，p），为（$p-1$）维超平面的约束条件。至此就完成了第一约束条件的降维处理。

（3）浓度非负约束（第二约束，INNPOL）

从式（6-43）可知，任一该体系的混合物波谱 x_i^t 在正交的抽象波谱空间 $\{q_1^t$，q_2^t，…，$q_p^t\}$ 中可由点集（u_{i1}，u_{i2}，…，u_{ip}），（$i=1$，2，…m）来表示，再考虑式（6-44a），因 u_{i1}是（u_{i2}，…，u_{ip}）的线性组合，只要（u_{i2}，…，u_{ip}）已知，u_{i1}就可唯一确定，亦即混合物波谱 x_i^t 实际就可由（u_{i2}，…，u_{ip}）构成的（$p-1$）维超平面唯一确定。由混合物波谱 x_i^t 点集（u_{i2}，…，u_{ip}）（$i=1$，2，…m）在（$p-1$）维超平面所围成的区域，就构成了浓度非负的约束区域，Borgen 等将其称为 INNPOL。

如采用这一统一降维技术，就可使二组分混合体系的可行解域落在一条直线之上，而使三组分混合体系的可行解域落在一二维平面之上。

对于二组分混合体系，根据式 $1+\sum_{i=2}^{n}f_{ki}t_{jk}\geqslant 0$，有

$$1+f_{2i}t_{j2}\geqslant 0$$

亦即由

$$t_{j2}\geqslant -1/f_{2i}\qquad (j=1,\ 2)$$

来确定，这几条直线就构成了波谱数值非负的约束条件，而 m 个混合物波谱也可由 u_{i2}（$i=1$，2，…，m）来表示出。这样，在本章第二节讨论的图 6-22 变成一条直线，可由图 6-30 示意表示出。

对于三组分混合体系，式（6-288）变成

$$1+f_{2i}t_{j2}+f_{3i}t_{j3}\geqslant 0\qquad (i=1,2,\cdots,n) \tag{6-289}$$

即由式（6-290）表出的 n 条直线

$$1+f_{2i}t_{j2}+f_{3i}t_{j3}=0\qquad (i=1,2,\cdots,n) \tag{6-290}$$

构成第一约束条件的区域（FIRPOL），而 m 个混合物波谱向量则可由 m 个点集（u_{i2}，u_{i3}）（$i=1$，2，…，m）来表出，由此可得到第二约束条件的区域（INNPOL），图 6-31 示意地表出了这一情况。

从上述讨论可知，通过如此降维处理，可将 p 组分体系降至（$p-1$）维空间来进行图像

表征，以获得 p 个组分的可行解域。Borgen 等还对四组分混合体系的情况进行了讨论[illegible]，得到了三维的可行域图像，但是，计算和相应的图像表征也变得更加复杂。值得提出的是，如果体系的组分数大于 5 时，即使经如此降维处理亦难用图像表征出来；而且，即使求得了体系纯组分光谱的可行解域，仍未得到具有真实物理意义的唯一解。可以想像，如果遵循这一思路继续研究下去，恐怕一时还难于得到可为广大分析工作者易于接受、易于理解且又令人振奋的结果，似有必要开辟新的研究途径。

随着分析实验室中的色谱联用仪器的出现和普及，使这样的设想成为可能。下面我们将针对近年来专门针对二维色谱联用仪所产生的二维数据进行矩阵分辨的方法给出详细的介绍。可以这么说，以下将要介绍的黑色分析体系的矩阵分辨解析方法实际上都是基于本节所讨论的自模式曲线分辨法，唯一不同的地方是在于它们更多地使用了二维色谱联用仪所产生的二维数据的特点，其中最重要的是色谱的分离特性，而不只是局限于浓度非负、光谱非负的约束，使得直接快速解析复杂未知的黑色分析体系出现了新局面，同时，由于使用了二维色谱联用仪所产生的二维数据的其他化学特点，也使得获得具有真实物理意义的唯一解成为可能，取得了令人振奋的新结果，为直接快速的解析复杂未知的黑色分析体系提供了一条崭新途径。

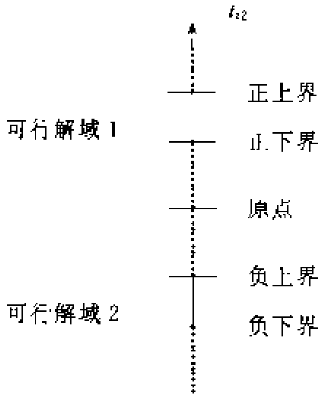


图 6-30 二组分体系约束界示意图

……禁止界；——可行解域

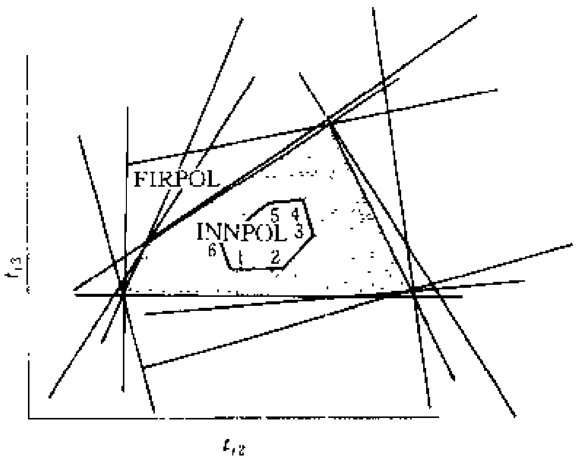


图 6-31 三组分含 6 个混合物波谱的纯组分波谱可行解域确定示意图

阴影部分为纯组分光谱的可行解域

（二）迭代目标转换因子分析法

【基本思路】

迭代目标转换因子分析法（ITTFA）由 Gamperline 和 Vanteginste 分别独立提出[152,153]，其主要思路是利用主成分分解的抽象矩阵来进行迭代，以求得一个逼近的可能解。实因将 $\boldsymbol{X}$ 矩阵分解后，可以得到两个抽象正交矩阵 $\boldsymbol{T}$ 和 $\boldsymbol{P}^t$，分别构成了正交且又与纯浓度谱和纯光谱张成的空间为同一线性空间，这样，用抽象正交空间来替代纯浓度谱和纯光谱张成的空间，对一初始矢量来进行迭代逼近，以求得一个逼近可能解。其实，由 Gemperline 和 Vanteginste 分别提出迭代目标转换因子分析法之区别仅在于他们所取的初始迭代矢量不同。

【基本算法】

设已通过某种量测方法获得了一个二维数据矩阵 $\boldsymbol{X}$，并假设此二维数据矩阵是双线性的，即 $\boldsymbol{X}$ 的每一列都表示一个浓度分布（对于色谱数据就是色谱），每一行都表示一个混合光谱。采用主成分分析方法先对此二维数据矩阵 $\boldsymbol{X}$ 进行分解，于是得到

$$\boldsymbol{X}=\boldsymbol{T}\boldsymbol{P}^t+\boldsymbol{E} \tag{6-291}$$

式中，T 和 P^t 为两个正交矩阵，其中 T 是列正交矩阵，而 P^t 为一行正交矩阵，有时也把它们分别称之为抽象色谱阵和抽象光谱阵；E 为残差矩阵。另外，根据 Lambert-Beer 定律可知，

$$X=CS^t+E \tag{6-292}$$

式中，C 为（$m\times n$）阶的真实浓度分布阵（对于色谱数据就是色谱），S^t 为（$n\times l$）阶的真实纯物种光谱阵。在这里 m 为色谱在保留时间轴上的采样点数，l 为光谱的波长点数，n 为混合体系的独立组分数。对照上述式（6-291）和式（6-292）可以看到

$$CS^t=TP^t=TRR^{-1}P^t \tag{6-293}$$

即

$$C=TR \tag{6-294a}$$

和

$$S^t=R^{-1}P^t \tag{6-294b}$$

式中，R 为一 $n\times n$ 的满秩旋转矩阵，它可以将抽象浓度和抽象光谱阵转换成为有物理意义的、真实的浓度分布阵和真实的纯物种光谱阵，而这正是所要求的。

很多基于自模式曲线分辨方法的算法，其目的也就在于找到这个转换矩阵 R。如果将式（6-294a）（在此只讨论浓度分布阵的情况，对于纯光谱阵如式（6-294b）所示出的情况与此类似，可以照推）写成矢量式，即

$$c_i=Tr_i \tag{6-295}$$

式中，下标 i 表示真实的浓度分布阵的第 i 列。设已通过某一方法得到了 c_i，就可以很容易地采用最小二乘方法求出转换矩阵的第 i 列来

$$r_i=(T^tT)^{-1}T^tc_i \tag{6-296}$$

由于 T 为一列正交的矩阵，所以（T^tT）的求逆很容易。在此的问题是：c_i 其实是未知的，是我们最终要求得的。如前所述，迭代目标转换因于分析（ITTFA）的主要思路是利用迭代来解决这一矛盾。先找到一个初始迭代矢量 c_i^0，由式（6-296）算出一个相应的 r_i^0，再由式（6-295）就可得到一个新的浓度分布矢量，记为 c_i^1，就此循环使用上述两式，直到 $\|c_i^{k+1}-c_i^k\|$ 小于某一给定常数，即停止迭代，由此而得到的 c_i^{k+1} 也就是我们要求的可能解。众所周知，由迭代法所求解时，初始迭代矢量的选择具有决定性意义。如前所述，由 Gemperline 和 Vanteginste 分别提出迭代目标转换因子分析法之区别就在子他们所取的初始迭代矢量不同，在此只介绍 Gemperline 的方法，对 Vanteginste 的方法不详细介绍，只因他是采用因子分析中的最大方差旋转法来寻找初始迭代矢量，较为复杂，且物理意义不明确，有兴趣的读者可参阅文献[153]。Gamperline 的方法主要基于如下事实，即利用纯物种的浓度分布(尤其是对色谱而言)有一个且只有一个极大值的特点，所以他选择的初始迭代矢量为 $c_i^0=(00\cdots0100\cdots0)^t$，即一个所含元素皆为零，其中只有一个元素是 1 的矢量为其初始迭代矢量，以引导迭代朝某一纯物种的浓度分布收敛。此外，在迭代过程中，每得到一个迭代的浓度分布都要进行修正，即利用浓度分布中的每一元素不可能为负数的条件，把迭代所得的浓度分布矢量中的负元素全部置成零，然后再继续迭代。这一步相当重要，它可促进迭代朝真实纯物种浓度分布逼近。

【讨论】

对于纯物种浓度分布的重叠程度不严重的混合体系，该法的确可以收敛至真实的纯物种浓度分布[154]，但当纯物种浓度分布的重叠程度严重时，一般难于得到真实解，只可能是一种可能解。当然，怎样找到更有利的初始迭代矢量仍是一个值得研究的问题，本书作者在此方面亦做过一些探讨，有兴趣的读者可参阅文献[155]。

在此，分析一下该法的物理意义，对正确使用该法和理解该法的适用性有帮助。该法的

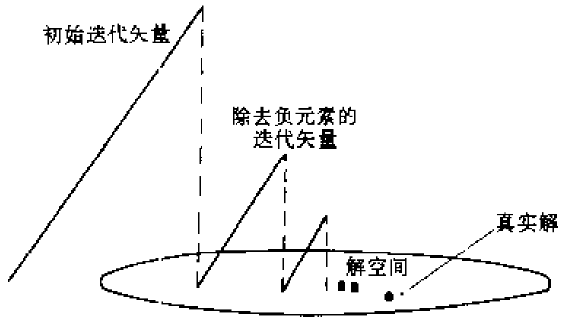


图 6-32 迭代目标转换因子分析法的迭代逼近几何示意图

基础在于它利用了两个抽象正交矩阵 $\boldsymbol{T}$ 和 $\boldsymbol{P}^t$ 分别构成与纯浓度谱阵 $\boldsymbol{C}$ 和纯光谱 $\boldsymbol{S}^t$ 张成的空间为同一线性空间的事实，所以，用正交矩阵 $\boldsymbol{T}$ 来代替纯浓度谱阵 $\boldsymbol{C}$ 以进行迭代逼近，是可能求得一个逼近可能解的。整个迭代过程示意地由图 6-32 表出，以帮助读者对该法的理解。

（三）渐进因子分析及其相关方法

【基本思路】

渐进因子分析法的主要思路在于它较精彩地利用了二维数据的一个重要化学特点，即化学物质的在很多情况中的依次出现特性，具有这样特性的二维数据包括联用色谱（如 GC-IR，HPLC-DAD，GC-MS 等）数据、滴定光谱数据（如在不同 pH 值的光谱）、动力学光谱数据等。渐进因子分析法最初由 Zubbuhler 和 Maeder 等提出[156]，主要用来解决光谱滴定数据的问题，后由 Maeder 进一步拓展[157~159]，可用于色谱分辨。渐进因子分析法的思路很简单，它利用主成分分析所得的特征值为其基本出发点，不同之处只是逐步渐进地延拓至整个矩阵，从而得到各物种在该数据矩阵中的出现点和消失点。

【基本算法】

其具体算法将通过图 6-33 来解释。从图 6-32 可以看出，它的计算是由两大部分组成，第一部分叫前向计算，从矩阵的前两行开始（这是因为进行主成分分析至少要两行），继采用逐步增大计算的矩阵来进行，即首先计算只含有两行的子矩阵的主成分，可得两个特征值，然后计算含有三行的子矩阵的主成分，得三个特征值，仿此依次增大子矩阵的行数，直至计算完整个矩阵；第二部分叫反向计算，从矩阵的最后两行开始，仿第一部分的思路然后计算含有三行的子矩阵的主成分，依次增大子矩阵的行数，直至计算完整个矩阵。这样就得到一系列的特征值，如果体系的组分数为 n，就把（$n+4$）或者是（$n+5$）个大的特征值的对数对保留时间作图，而得到各物种在该数据矩阵中的出现点和消失点。当然，在做此分析之前，组分数为 n 一般是未知的，所以一般是多取几个，以免丢失有用信息。

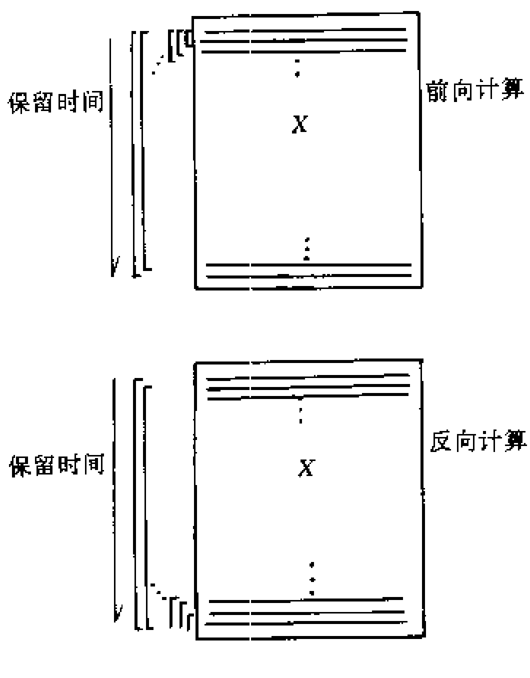


图 6-33 渐进因子分析法示意图

仍以联用色谱的二维数据为例，在图 6-34 中示出了一个三组分体系的渐进因子分析的结果图，从图 6-34 可以看出，渐进因子分析的结果在理想的情况下是确实能为我们确定体系物种数，及确定各物种在该数据矩阵中的出现点和消失点提供重要信息的。以后还可以看到，有了这样的信息，将对下一步的纯光谱和纯色谱的分辨提供很大方便。

【讨论】

渐进因子分析法虽能给出这些有用的信息，但是它的计算量是比较大的，如要完成一个 100×60 阶的矩阵的计算，大约就需要几个小时，实因随着矩阵的增大，主成分分析所需时间

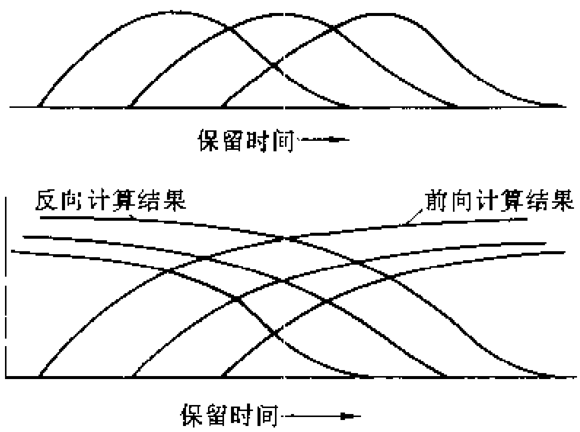


图 6-34　渐进因子分析法结果示意图

增加很多，而且，它的计算是分两个方向进行的，是不是存在一种方法，它既可得到渐进因子分析法所能得到的信息，又能在几分钟之内完成呢?答案是肯定的，这就是将要介绍的固定尺寸移动窗口渐进因子分析法。

固定尺寸移动窗口渐进因子分析法是由Keller 和 Massart 提出[160]。其思路相当简单，与渐进因子分析法逐步扩大主成分分析窗口的做法不同，它却固定一个窗口的尺寸，只是不断将此窗口从二维数据的起点移向终点，从而得到一个特征值的对数值对保留时间的作图，以得到二维色谱数据在保留时间方向的信息。该法的基本算法和所得到的信息将图示于图 6-35。从图 6-35 (a) 可以看出，该法的计算过程为：① 先选择一个窗口的尺寸，一般为 7，即选择 7 个行矢量组成一个于矩阵；② 用这样的尺寸来做主成分分析，可得 7 个特征值；③ 如图 6-35 (a)所示，依次移动这个窗口，如先算 1～7 行所组成的子矩阵，然后，计算 2～8 行所组成的子矩阵，…，直到计算到最后7 行所组成的子矩阵，这样就可以得到一含 7 个特征值的系列，将其对相应的保留时间数据点作图将得到如图 6-35 (b) 所示的结果图。Keller 和 Massart 提出该法的初衷主要是要进行二维色谱峰的纯度检验，就如图 6-35 (b) 所示那样，如果该色谱峰不纯，在 FSMFEFA 的特征值对数图中就会出现一个小峰，这个小峰的存在就说明存在着另一种化学物质，实因在此图中第一个特征值表示主要化学物质的存在，而第二个特征值大于噪声就表示在它出现的区域还存在着第二个物种，否则，这个色谱峰始终只会有第一个特征值很大，而其余的特征值都接近于噪声。

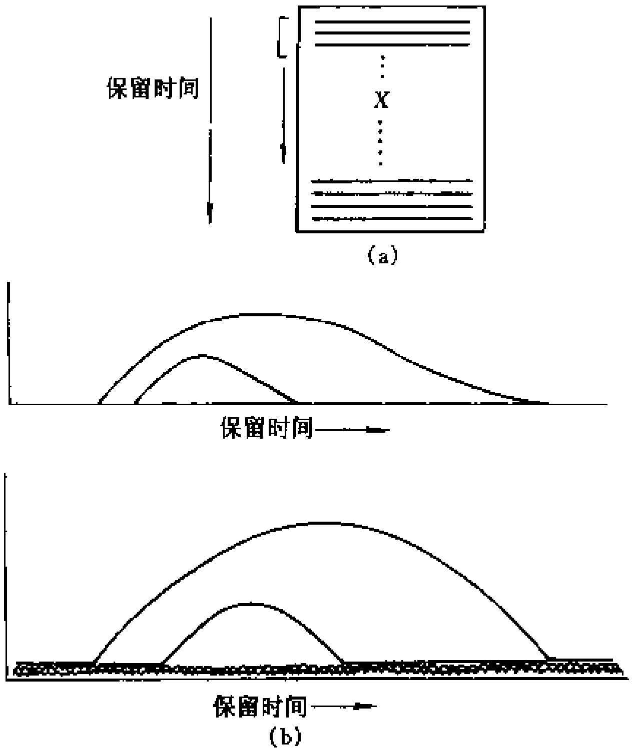


图 6-35　固定尺度移动窗口渐进因子分析法示意图
(a) 方法的主要思路示意图；(b) 方法的结果示意图

然而，在使用这一技术来进行二维色谱峰纯度检验时，有一点是特别值得注意的，即所谓不等性方差噪声的影响问题，由于这种噪声的存在，常常不能得到正确的结论，具体的例子可参见文献[159]。由子在实际量测数据中经常存在不等性方差噪声的情况，即量测噪声随分析信号大小而变化，例如在紫外可见光谱的量测中，量测噪声的绝对值在一定范围中随量测信号而增大，但量测的相对误差却相对稳定，Keller 和 Massart 对此在 HPLC-DAD 仪器上进行实验研究的结果证实了这一结论。注意到主成分分析有关判别信号和噪声的区别是基于特征值的绝对值的大小，而且主成分分析模型也是建立在等性方差噪声的假设之上的，所以，在有这种不等性方差噪声存在时，主成分分析就很难奏效了。为此提出了一个

对二维色谱数据的一种预处理方法，以克服不等性方差噪声所造成的困难。此预处理方法的基本思路很简单，即对那些处于色谱峰的光谱（就是那些光谱积分值大于某一给定值α时）进行降幅处理，而这种处理又不影响数据原有的线性结构，不至于影响下一步的主成分分析，它可由式（6-297）表示出。

$$x'_{ij}=(x_{ij}/\sum x_{ij})\alpha \qquad (\sum x_{ij}>\alpha) \qquad (6\text{-}297)$$

式中，x_{ij}为原始二维数据矩阵中第i行第j列的元素，而x'_{ij}则为处理后的数据。

在进行了这样预处理后的数据x'_{ij}的光谱积分就为α。经这样处理后，不等性方差噪声的就可以消除了。图6-36示出了这样的实例。经这样的预处理，我们对一药物体系进行纯度检验的结果表明，用FSMWEFA可检出0.7%的异构不纯物。详细的讨论请参看文献[161]。

还有一点在此值得提出的是，实际上FSMWEFA不只限于二维色谱峰的纯度检验，它也可像渐进因子分析法那样用于一般的多组分体系，这在本章第五节还会详细地进行讨论。这样的一种改进看起来很一般，可是，它却把整个计算的时间大大缩短，实因它的主成分分析（就用MATLAB软件所提供奇异值分解算法可直接计算）计算始终是针对一个很小的子矩阵，这样，完成一个二维色谱数据的计算一般只需要几分钟，从以后的讨论我们还可以看到，它不但可以得到和渐进因子分析法（EFA）同样的信息，而且从某种角度来说，是更准确、更可靠的信息，实因该法实质上用的是一种局部主成分分析的方法。关于为什么局部主成分分析要优于全局主成分分析方法（即前一节中讨论的主成分分析法），以后还会进行详细讨论。

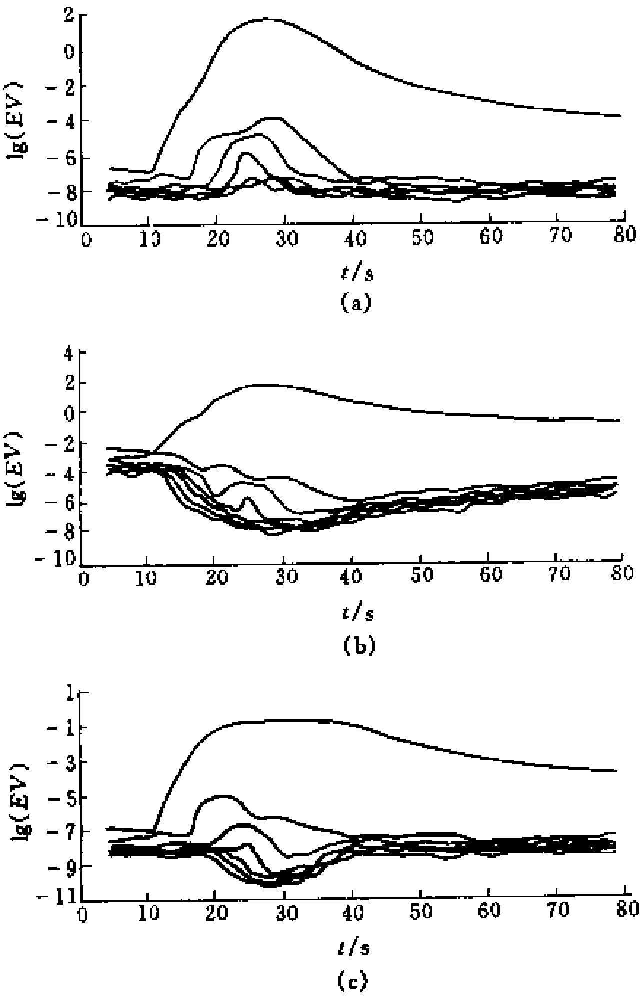


图6-36 两个药物异构体混合体系（两色谱峰的分离度为0.5；含量少的组分为含量高组分的0.7%）之特征值对保留时间变化图

(a) 原始数据的特征值的对数对保留时间变化图，从图可看出有四个特征值大于基线水平；
(b) 数据经平方根转换处理后的特征值的对数对保留时间变化图，从图看出只有一个特征值大于基线水平；
(c) 数据经式（6-297）处理后的特征值的对数对保留时间变化图，从图可看出只有两个特征值大于基线水平，可检测出0.7%的药物异构体

（四）窗口因子分析法

【基本思路】

窗口因子分析法（WFA）由Malinowski提出[162]，其主要思路是利用一个只有$(n-1)$个物种的“窗口”，从而求出那第n个物种的纯色谱来，此思路可由图6-37简要示出。

为便于理解，在此仍采用前几节所采用的符号，即二维数据矩阵为$\boldsymbol{X}$，用主成分分析方法对此二维数据矩阵$\boldsymbol{X}$进行分解可得到

$$X=TP^t+E=\sum_{i=1}^{n}t_ip_i^t+E \quad (6\text{-}298)$$

另外，根据 Lambert-Beer 定律可知，

$$X=CS^t+E=\sum_{i=1}^{n}c_is_i^t+E \quad (6\text{-}299)$$

式中，n 为体系的组分数。因为 T 和 C、P^t 和 S^t 张成同一线性空间，所以，有

$$c_i=\sum_{j=1}^{n}\beta_{ij}t_j \quad (6\text{-}300a)$$

或

$$s_i^t=\sum_{j=1}^{n}\alpha_{ij}p_j^t \quad (6\text{-}300b)$$

式中，β_{ij}和 α_{ij}分别为线性组合系数。

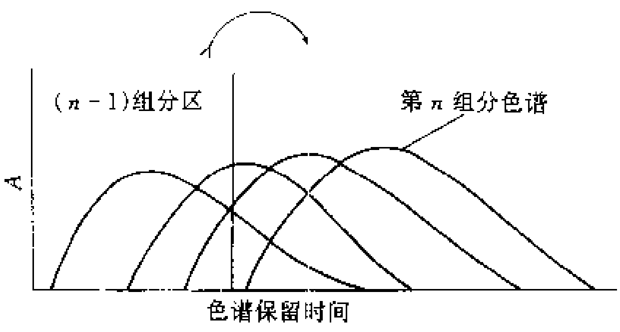


图 6-37　窗口因子分析法主要思路示意图

现设 X 为只含（$n-1$）个组分的二维数据矩阵 X 的一个子矩阵，为方便起见，就为只含了前（$n-1$）组分，经主成分分析可得到（$n-1$）正交矢量，即

$$X=T^0P^{0t}+E=\sum_{i=1}^{n-1}t_i^0p_i^{0t}+E \quad (6\text{-}301)$$

式中，用上标 0 来表示这些正交矢量，是说明 t_i^0 与 t_i 不同。同式（6-300），有

$$c_i=\sum_{j=1}^{n-1}\eta_{ij}t_j^0 \quad 或 \quad s_i^t=\sum_{i=1}^{n-1}\gamma_{ij}p_j^{0t} \quad (6\text{-}302)$$

在此与式（6-300）不同的是，式（6-302）表出的只是一个（$n-1$）维的 X 的子空间，即只能将那包括在此区域内的（$n-1$）物种。事实上，式（6-302）可以拓广至 n 维空间，即通过找到一个与 p_j^0（$j=1$，…，$n-1$）正交的矢量 p_n^0，则有

$$p_i^t=\sum_{i=1}^{n}\kappa_{ij}p_j^{0t} \quad (6\text{-}303)$$

因为 p_j^{0t} 的正交性，κ_{ij}很容易求出，即

$$\kappa_{ij}=p_i^tp_j^0 \quad (6\text{-}304)$$

问题是如何来求出 p_n^0？对式（6-303）中的 p_i^t 求和，则有

$$\sum_{i=1}^{n}p_i^t=\sum_{i=1}^{n}\left(\sum_{j=1}^{n}\kappa_{ij}p_j^{0t}\right)=\sum_{j=1}^{n}\left(\sum_{i=1}^{n}\kappa_{ij}\right)p_j^{0t} \quad (6\text{-}305)$$

对式（6-305）进行重写，可得

$$\sum_{i=1}^{n}p_i^t=\sum_{i=1}^{n}\left(\sum_{j=1}^{n}\kappa_{ij}p_j^{0t}\right)=\sum_{j=1}^{n-1}\left(\sum_{i=1}^{n}\kappa_{ij}\right)p_j^{0t}+\left(\sum_{i=1}^{n}\kappa_{in}\right)p_n^{0t} \quad (6\text{-}306)$$

于是有

$$\left(\sum_{i=1}^{n}\kappa_{in}\right)p_n^{0t}=\sum_{i=1}^{n}p_i^t-\sum_{j=1}^{n-1}\left(\sum_{i=1}^{n}\kappa_{ij}\right)p_j^{0t} \quad (6\text{-}307)$$

实际上，式（6-307）就给出了一个求得 p_n^{0t} 的方法，实因该式右边的所有量都为已知量或是可求出量，而左边的（$\sum\kappa_{in}$）则只是一个正规化常量。

这样得到的由（$n-1$）维子空间拓广而得的正交空间 p_j^{0t}（$j=1$，…，n），因其与原始的 n 维正交空间 p_j^t（$j=1$，…，n）可以相互线性表出，所以它们实际上张成同一空间。于是，无论是采用 p_j^{0t}（$j=1$，…，n）还是 p_j^t（$j=1$，…，n）来表示该体系的纯物种光谱都是可行的。所以有

$$s_i^{t}=\sum_{j=1}^{n}\gamma_{ij}p_j^{0t} \tag{6-308}$$

将式（6-308）代入式 $X=\sum_{i=1}^{n}c_i s_i^t$ 可得

$$X=\sum_{i=1}^{n}c_i\left(\sum_{j=1}^{n}\gamma_{ij}p_j^{0t}\right)=\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}c_i p_j^{0t} \tag{6-309}$$

在式(6-309)中已将量测误差矩阵项忽略。对式(6-309)两边左乘 p_n^0，则有

$$Xp_n^0=\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}c_i p_j^{0t}\right)p_n^0 \tag{6-310}$$

由于 p_n^0 与 $p_j^0(j=1,\cdots,n-1)$都正交，即 $p_j^{0t}p_n^0=0(j=1,\cdots,n-1)$，所以有

$$Xp_n^0=\left(\sum_{i=1}^{n}\gamma_{in}c_i\right)p_n^{0t}p_n^0 \tag{6-311}$$

因为当 s_i^t 中的 i 小于 n 时，即纯物种被包括在那$(n-1)$个之中时，s_i^t 都可以由前$(n-1)$个 p_j^{0t} 来线性表出[参看(6-302)式]，此时就有 γ_{in}对除 n 外的所有 i 来说都等于零，只有 s_n^t 需要 p_n^0 的线性组合得到，同时是可以归一化的，这样

$$Xp_n^0=\gamma_{nn}c_n \tag{6-312}$$

即将归一化后的 p_n^0 左乘整个矩阵 X 时，就可以得到 c_n 与一常数因于 γ_{nn}的乘积，也就是说，得到了纯物种 n 的纯色谱，将其归一化即可得到归一化的色谱了。仿此对其他$(n-1)$个物种同样处理，只要可以找到一个只含不包括要找的色谱在内的其他$(n-1)$个物种存在的区域，或称窗口，就可以将这 n 组分的体系进行唯一的分辨结果。

【基本算法】

从上述讨论可知，窗口因于分析法（WFA）的具体算法可由下述几步组成：

① 确定只含（$n-1$）个组分的二维数据矩阵 X 的一个子矩阵。一般可用前节讨论的固定尺寸移动窗口渐进因于分析法或特征结构跟踪分析，更准确地得到在保留时间方向上的化学秩图来确定。

② 用式（6-307），即

$$\left(\sum_{i=1}^{n}\kappa_{in}\right)p_n^{0t}=\sum_{i=1}^{n}p_i^t-\sum_{j=1}^{n}\left(\sum_{i=1}^{n}\kappa_{ij}\right)p_j^{0t}$$

求得 p_n^{0t}；

③ 用式(6-312)，即

$$Xp_n^0=\gamma_{nn}c_n$$

即可求得 c_n 来。

④ 重复①～③步，直至求得所有的 c_i（$i=1, 2, \cdots, n$），继用最小二乘法求得各纯物质光谱，即

$$S^t=(C^tC)^{-1}C^tX$$

Malinowski 将 WFA 对 Gemperline 和 Hamilton 所做的一套流动注射分析的有关七种不同铋络合离子数据[163]进行了分辨，得到满意的结果，有兴趣的读者可参阅文献[162]。

（五）直观推导式演进特征投影法

从前几节的讨论中可以看到，迭代目标转换因于分析法（ITTFA）、渐进因于分析法（EFA）、固定尺寸移动窗口渐进因于分析法（FSMWEFA）以及窗口因子分析法（WFA）这些方法都是在利用二维色谱的某些化学特征来进行解析，实际上它们都没有完全利用二维色

谱数据的化学特征，而直观推导式演进特征投影法（HELP）方法则旨在充分利用二维色谱数据的所有化学特征。所以，我们在讨论 HELP 方法之前，似有必要对二维色谱数据的所有化学特征进行剖析，以便于读者对该法的理解。

因二维色谱数据具有以下对分析化学特别有用的 6 个主要化学特征，所以它可为分析化学家们提供机会，从而使得用二维色谱数据解析最困难的黑色分析体系成为可能。以下将对它们一一加以详细讨论。

① 二维色谱数据同时包含了光谱与色谱信息。一般说来光谱（如质谱、红外光谱、紫外可见光谱等）能提供化学物质的大量定性信息，而色谱则能提供定量信息，如果条件控制得好，化学物质的保留时间对化学物质的定性也是很有帮助的，这说明二维数据与通常的一维数据相比，可多提供很多新信息，为分析化学家提供了新的机遇，也就是说，我们可以同时在光谱或色谱空间来研究我们的样本，为我们直接用这样的仪器来解析复杂的多组分分析化学体系提供了机会。

② 色谱仪器具有很强的分离能力，它可将一个很复杂的分析体系分解成为多个相对简单的子体系，这样，也就为分析化学工作者提供一种很有用的将复杂体系变为简单体系的工具和思路，使我们有可能只需直接处理这些相对简单的体系或子矩阵，而毋需直接用数学方法来解析复杂的多组分分析化学体系。

③ 纯化学物种的色谱峰有一个也只有一个最大峰值，这一化学数据的特点，可为我们在进行数据处理和分析时带来有用信息。

④ 化学物质在色谱仪中流出时，不同的化学物质是依照某种化学规律依次流出和依次消失的，这一特点，就像我们在前几节的讨论的那样，为确定某一物质的色谱保留时间存在区间带来很大方便。

⑤ 在二维色谱数据中，经常很容易找到所谓的零组分区，即在一段保留时间区域内没有任何化学物质流出，正是这样的零组分区域，为我们直接提供了在这一样本运行条件下的仪器噪声或仪器背景的有用信息，这一信息将为我们确定体系的化学组分数，即将分析信号与噪声的分离和区别，提供重要信息，此外，如果体系存在某种背景，零组分区域的信息，也将直接为我们提供有用的仪器背景信息。

⑥ 一般二维色谱数据都是非负实数，像我们在本章第一节中看到的，这一条件将为我们提供很有用的约束条件，从而大大缩小了搜索空间。

有了这样的前提，再来看一看黑色分析体系的数学模型，即

$$\boldsymbol{X}=\boldsymbol{C}\boldsymbol{S}^{t}+\boldsymbol{E}$$

在此我们所知的只有二维数据量测矩阵 $\boldsymbol{X}$，而对于 $\boldsymbol{C}$ 和 $\boldsymbol{S}^{t}$ 都一无所知，此外，就是对 $\boldsymbol{C}$ 和 $\boldsymbol{S}^{t}$ 中各有多少行与多少列，即 n 为几也是未知的。这样的问题在数学上称为矩阵反问题，一般说来是不可能存在唯一解的，除非预先有一定特殊条件，这在讨论自模式曲线分辨方法时就已经看到，如果只用到浓度非负和波谱量测值非负的条件，最多就只能得到可能解。从这一角度来看，上述提出的二维色谱数据的化学特征就显得十分重要了。

直观推导式演进特征投影法（HELP）由本书作者与 Kvalheim 共同提出[164,165]，它是采用一种逐步的推导式方式来解析二维色谱数据，反自模式曲线分辨法仅基于全局主成分分析的思路，它主要基于局部主成分分析[166]，以尽量利用色谱的分离特性来获得各子矩阵的有用信息，其中最重要的是所谓选择性信息，在此基础上，它试图根据色谱流出数据特性来获得在色谱保留时间方向上的所谓"秩图"，实际上就是色谱保留时间方向上组分分布图；然后根据

秩图和所得的选择性信息来对重叠的色谱或光谱进行满秩分解，以获得唯一解。以下将对HELP方法提出的一些基本概念和解析手段作出较详细的讨论。

【基本概念与基本过程】

1. 演进特征投影图和选择性信息

所谓演进特征投影图实际就是主成分投影图。首先，对二维色谱数据进行主成分分析（或就是进行奇异值分解分析），然后从两个空间，即所谓的色谱空间和光谱空间，用第一主成分对第二主成分作图，这样就可以得到很多的有用信息。在讨论这些有用信息之前，有必要首先来讨论一下二维色谱数据的特点以及此时采用特征投影图的矢量代数的基础和整体优越性。二维色谱数据实际上是由两个高维的坐标，即光谱的波长空间坐标和色谱的保留时间空间坐标来决定，联用色谱二维数据的每一个光谱点，即二维数据中的每一行，都可由波长坐标空间的一个矢量来决定，不同化学物质的光谱在此波长坐标空间都具有不同的位置。一般说来，任一高维空间的矢量可完全由其在该空间的方向和大小来唯一确定，这为我们用矢量来讨论二维色谱数据提供了很大方便，也就是说，只要知道了该矢量的大小和方向也就完全确定了该矢量。同理，对于联用色谱二维数据的每一个色谱点，即二维数据中的每一列，都可由保留时间坐标空间的一个矢量来决定，不同色谱在此保留时间坐标空间都具有不同的位置。这一情况可由图6-38形象地表出。这样，对于二维色谱数据，采用矢量来讨论它们之间的关系将有助于对其内在结构的理解，这也就是我们运用特征投影图来帮助解析二维色谱数据的初衷。

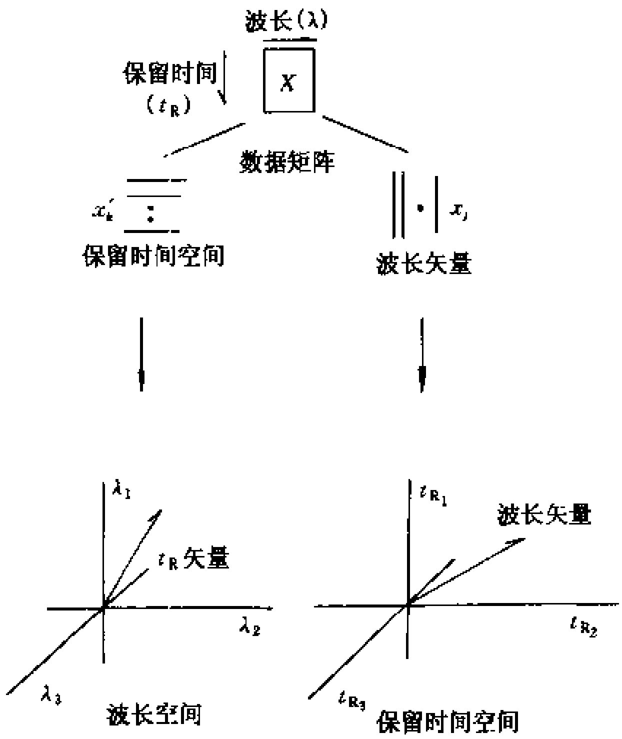


图 6-38 二维色谱数据在波长空间和色谱保留时间空间的矢量表示示意图

二维色谱数据矩阵中的行矢量都是波长空间中的矢量；二维色谱数据矩阵中的列矢量都是保留时间空间中的矢量

另外，在矢量空间中，一个矢量的决定因素在于其方向，即如果它的方向已经确定，它就可唯一地在此高维空间中被确定，或者用化学的语言来说，就是同一种化学物质，它的光谱在高维波长空间中就主要由其方向来确定，方向定了光谱也就唯一确定了，其大小只反映该物种的浓度。所以，如果该空间只存在一个化学物质，其光谱矢量在波长空间中就始终在一条通过原点的直线中变化，实因其方向始终不变。如果把这些矢量都看成是高维波长空间的一系列点列而连起来，它们将就是一条通过原点的直线段。注意到这样的结论对色谱保留时间空间也是同样适用的，所以，我们可以从两个方向来讨论二维色谱数据。图6-39和图6-40分别示出了在波长空间和色谱保留时间空间存在选择性信息的情况。

如前所述，所谓演进特征投影图实际就是主成分投影图，在此将就主成分投影图怎样反映上述高维空间的矢量特性来进行必要的说明。所谓主成分分析，实质上就是对二维数据进行奇异值分解（参见第十一章），即

$$X = UGP^{t} = TP^{t} \tag{6-313}$$

式中，U 为一列正交矩阵，称为得分矩阵，T 为 U 和 G 的乘积，实际上它的性质与 U 基本一致；G 为一对角矩阵，收集了 X 矩阵的特征值，如与前述特征值对应的话，是 X 矩阵的协方差阵所得特征值的平方根；P^{t} 为一行正交矩阵，也称为载荷矩阵。根据 P^{t} 的正交性对式（6-313）两边同时右乘 P，则可得

$$XP = T \quad 或 \quad x_i^{t}P = t_i^{t} \tag{6-314}$$

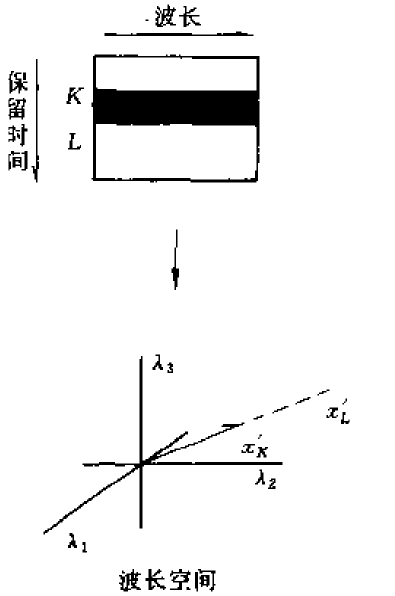


图 6-39　光谱波长空间的选择性信息

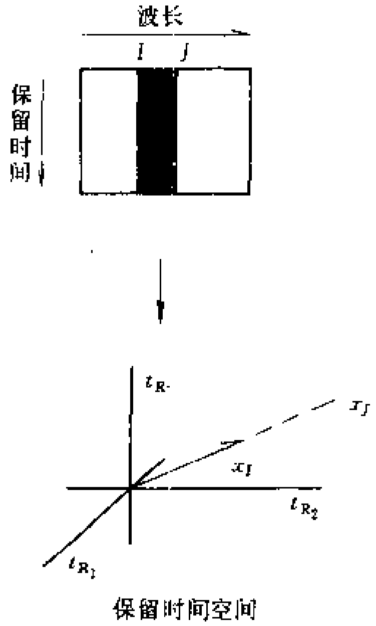


图 6-40　色谱保留时间空间的选择性信息

实际上，式（6-314）即表示将矩阵 X 的每一个行矢量 x_i^{t}（或者说每一个光谱）投影到正交载荷矩阵 P 之上，就得到了得分矩阵，所以得分矩阵实际上就反映了光谱之间的相互关系。同理，根据 T 的正交性对式（6-314）两边同时左乘 T^{t}，则可得

$$T^{t}X = P^{t} \quad 或 \quad T^{t}x_i = p_i \tag{6-315}$$

即表示将矩阵 X 的每一个列矢量 x_i（或者说每一个色谱）投影到正交得分矩阵 T^{t} 之上，亦即得到载荷矩阵，所以，载荷矩阵实际上就反映了色谱之间的相互关系。这样，就可以在低维空间来直接观察高维光谱和色谱之间的相互关系了。在演进特征投影图中，一般都只需取两个特征矢量来作图就够了，当然，如果能用三维的特征投影图应可得到更形象的信息。

在此特别提出，由于色谱的依次流出的性质，而使得二维色谱数据具有一个很重要的特点，即如果用曲线将这些投影点列连起来，我们就可以很容易地得到在二维色谱数据中的各化学物种依次流出的信息，这也是我们将其称为演进特征投影图的初衷。总的说来，利用演进特征投影图，可以很容易地得到以下有用信息：

① 色谱和光谱空间的选择性信息。特征投影图中经延长可通过原点的直线（基于最小二乘意义的），就代表色谱和光谱空间的选择性信息，用化学的语言来说就是所谓单组分区。得分投影图中显示的通过原点的直线段就表示在色谱（保留时间方向）空间中存在着选择性信息，亦即在保留时间的某一区域是纯组分区，从这里取出的光谱一定是某一种化学物质的纯光谱；同理，载荷投影图中显示的通过原点的直线段就表示在光谱（波长方向）空间中存在着选择性信息，亦即在波长的某一区域是纯组分区，从这些波长中取出的色谱一定是某一种化学物质的纯色谱。图 6-41 和图 6-42 分别示出了一个两组分体系和三组分体系的特征投影图的例子。这样的特征投影图很重要，据此可以很容易的找到色谱和光谱的选择性信息，也

就是说，在此充分地利用了二维色谱数据的分离性质。

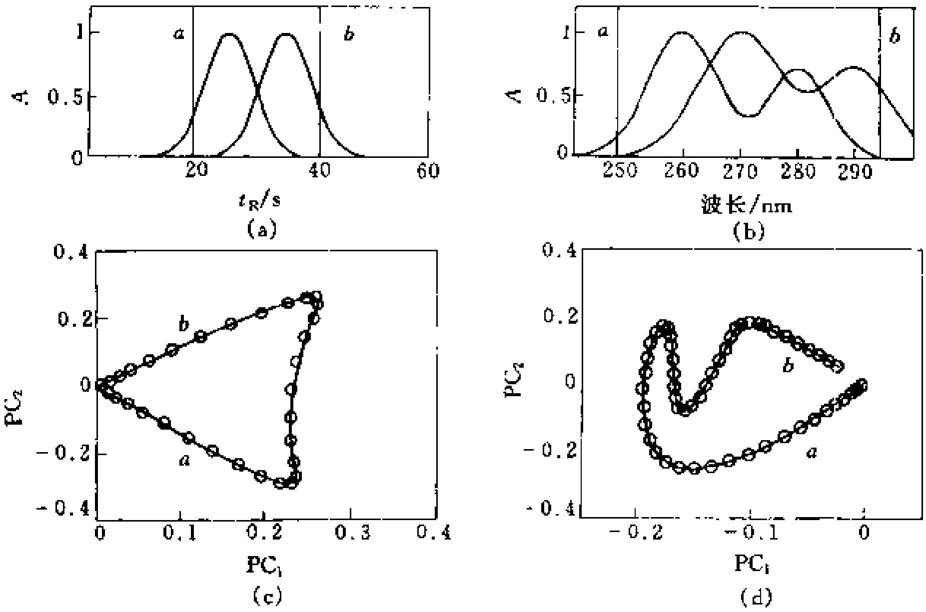


图 6-41 模拟 2 组分体系及其特征投影图

(a) 模拟体系两个纯组分色谱；(b) 模拟体系两个纯组分光谱；

(c) 波长空间的特征投影图；(d) 保留时间空间的特征投影图

图中标出的 a、b 部分为体系的选择性信息，从图中可以看

出，这样的选择性信息可由特征投影图的直线段中找出

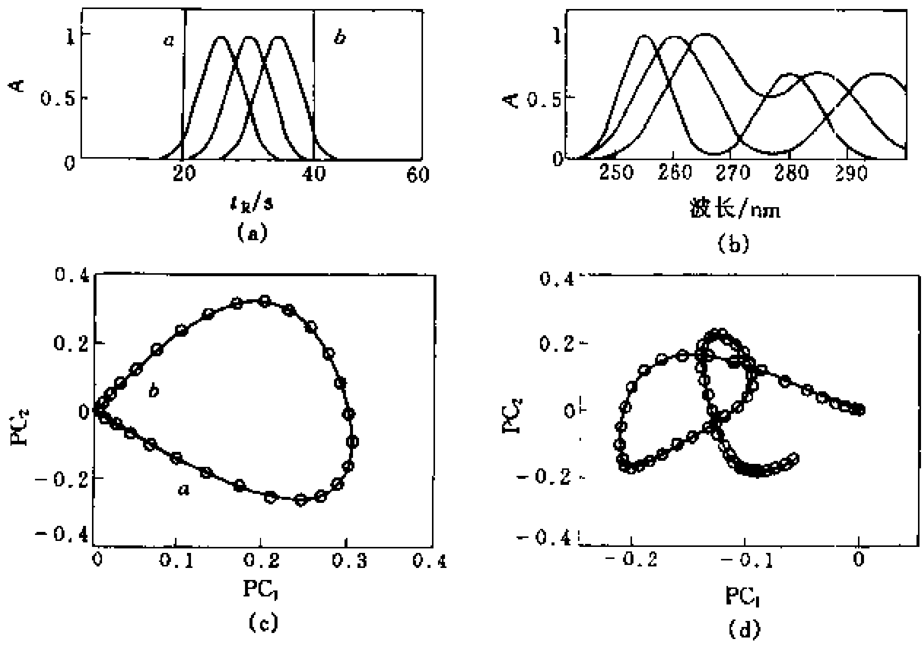


图 6-42 模拟 3 组分体系及其特征投影图

(a) 模拟体系三个纯组分色谱；(b) 模拟体系三个纯组分光谱；

(c) 波长空间的特征投影图；(d) 保留时间空间的特征投影图

图中标出的 a、b 部分为体系的选择性信息，从图中可以看出，与 2 组分体系相比，

3 组分体系因光谱重叠比 2 组分严重，已难于找到选择性信息

② 体系中各组分流出情况及其数目的确定信息。注意到不同的化学物种在演进特征投影图中都具有不同位置（或不同方向），所以，当一种新的化学物种出现的时候，如果它为第一

个，则为一从原点出发而不断增大的直线，而当此物种是第二或更后才流出来的，则会在演进特征投影图上与第一物种的点列形成各种曲线，从而为我们提供体系各物种依次流出的信息（参见图 6-41 和图 6-42）。这种信息不但对体系的组分数确定具有一分重要的意义，而且，它还可以用来区别不同的流出模式，这样的例子示于图 6-43，这也是为什么将此特征投影图称为演进特征投影图的原因。

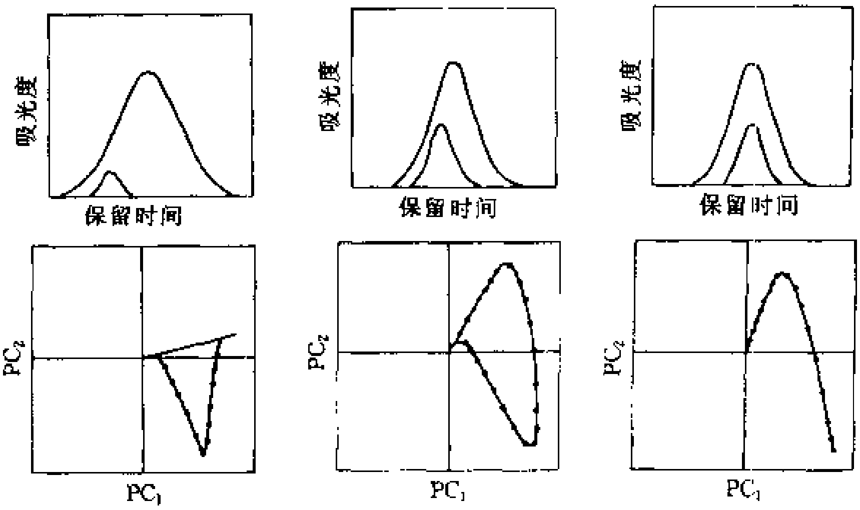


图 6-43　3 种不同的大色谱峰包小色谱峰之模式及其相应特征投影图

③ 二维色谱数据的背景信息。如果二维数据中不存在有背景，则零组分区的投影点应收敛于投影图中的原点，反之，如果零组分区的投影点不收敛于原点，则说明数据中存在着背景，在做进一步的分析之前必须想法扣除背景的影响（详见以后的讨论）。在图 6-44 中我们示出了这样的一个例子。这是因为如果零组分区的光谱为很小的服从正态分布的误差矢量，则其投影点相对子含有某种结构的光谱所代表的投影点一定在数值很小且随机地收敛于原点。所以，这样的特征投影图就为我们提供了一个方便的检测在二维色谱数据中是否存在背景的工具。

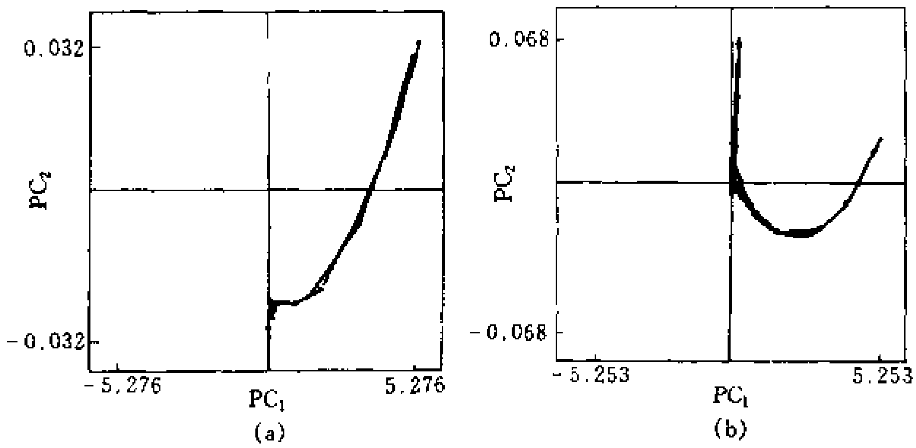


图 6-44　叶绿素 a 降解混合物的特征投影图

(a) 在背景扣除前的特征投影图（零组分区的光谱投影远离投影图原点）；

(b) 在背景扣除后的特征投影图（零组分区的光谱投影皆收敛于投影图原点）

2. 特征结构跟踪分析与秩图

特征结构跟踪分析[167]实际上是对固定尺寸移动窗口渐进因子分析法（FSMWEFA）的一种拓广。如前所述，FSMWEFA 可明显提高 EFA 的运算速度，且可同样提供我们所需要的各

组分在保留时间方向上的分布信息，但是，固定窗口的尺寸选取还存在着一些问题。当然，如果只将 FSMWEFA 用于色谱峰的纯度检验时，固定窗口的尺寸并不重要，但是如果想通过这一技术来获取各化学组分在保留时间方向上的分布信息的话，窗口大小的选取就十分重要了。实际上，可以这样认为，移动窗口的大小是与其检测的敏感度和选择性息息相关的，一般说来，如移动窗口尺寸大一些，它的检测第二组分的敏感度也相对高一些，而其检测单组分区（选择性区域）的选择件则越小。如一个单组分区仅含两个保留时间点，用包含三个保留时间的移动窗口就会将此有用的单组分区域漏掉。这样的例子示于图 6-45。如果说选择性信息对二维色谱数据的分辨具有很重要的意义的话（这一点将于以后给予详细讨论），那么从最小可能的窗口来使用这一技术就十分重要了。特征结构跟踪分析（ETA）的提出就正是建立在这一基础之上，它从窗口等于两个保留时间点开始进行移动计算，然后增大窗口尺寸为三，再进行一次移动计算，一般说来，只要窗口尺寸的大小与该体系中重叠峰中的组分数相当时，计算即可停止。在此，值得提出的是，移动窗口的大小选择还与色谱分辨率或色谱采样点的疏密有关，所以，窗口尺寸的大小还需根据具体数据情况而定，切不可简单使用一个尺寸，由此我们将可得到一个组分在保留时间方向上的分布信息图，称之为“化学秩图”，据此可将整个保留时间轴上的按重叠的化学组分数分成不同的区域。这样产生出来的秩图将示例于图 6-46。这样得到的秩图很重要，这是因为据此可以充分利用色谱的分离特性（二维色谱数据的第二特点）。

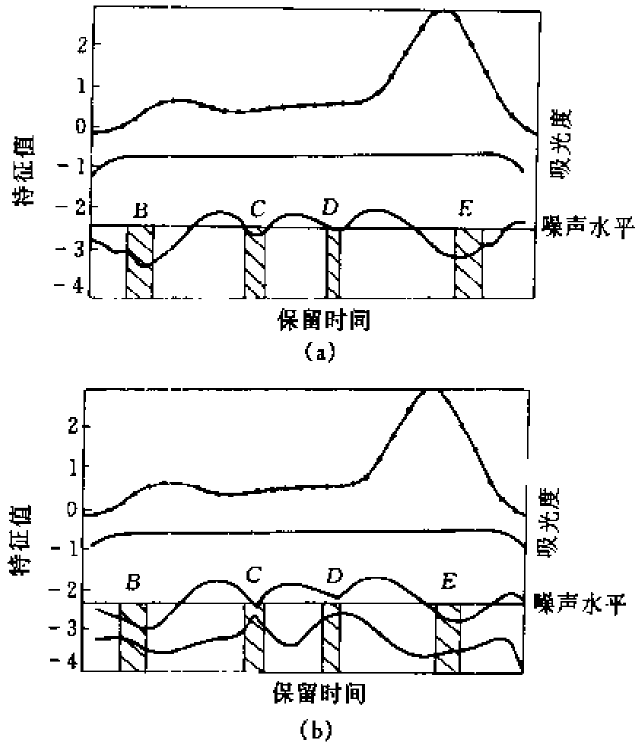


图 6-45　叶绿素 a 降解混合物的特征结构跟踪分析图

(a) 为窗口尺寸为 2 的跟踪分析图，从图中可见，B、C、D、E 皆为选择性区；(b) 为窗口尺寸为 3 的跟踪分析图，其中 D 区已明显不是选择性区域了

3. 零组分区和二维色谱数据的背景扣除

零组分区在二维色谱数据中十分重要，可是，这一重要的信息一般都被忽视，所以，我们认为有必要在这里特别提出来讨论。零组分区的重要意义可表现在以下两方面：第一如果在二维色谱数据不存在背景，则零组分区可直接提供就在这一次量测中的仪器噪声情况，这样直接提供的仪器量测噪声对用主成分分析中的主成分的正确确定将带来很大方便，这一点在本章第一节已说明。另外，即便是同一种仪器，不同的运行情况，噪声的特性亦有不同，所以如果我们对同一仪器总是假设它们具有同一噪声水平也是不严格的。从这一角度来看，二维色谱数据中零组分区提供的同一次量测中的仪器噪声情况就具有特殊意义了。怎样有效地利用零组分区的噪声信息来决定体系的噪声水平，在文献[168]中作出了较详细的讨论，在此不再赘述。第二，如果在二维色谱数据存在背景，零组分区则可直接提供就在这一次量测中的仪器所产生的背景情况，就利用这一信息可以对二维色谱数据的背景进行扣除。在此有必要指出的是，背景的存在将对主成分分析带来很大的影响，它不但影响体系组分数的正确确

定（如果是具有色谱漂移的背景一般可将体系的秩增加 2，有关详细讨论可参阅文献[169]），而且，它的存在对二维色谱数据的分辨亦影响很大。所以，在对二维色谱数据进行解析之前，如没有对存在的背景进行扣除，必将得到很差甚至是错误的结果。根据零组分区信息提出了一种二维色谱数据的背景扣除方法[170]，以下对该法作出简要介绍。

该法的基本点就是充分地利用零组分区的信息，实因在零组分区中无化学物质存在，所以，如果在零组分区存在某种具有规律性的信号，就实质上应该是反映了仪器的背景情况。该法的基本思路是比较一个色谱峰簇的流出前和流出后的局部主成分分析所得的载荷矢量（有时亦称抽象光谱），计算它们的相合系数，即此两个矢量的点积，如果它们的相合系数等于 1 或十分接近于 1 的话，就说明在此色谱峰附近存在同一的仪器背景，应该且能被有效的扣除。此思路可由图 6-46 示意表出。

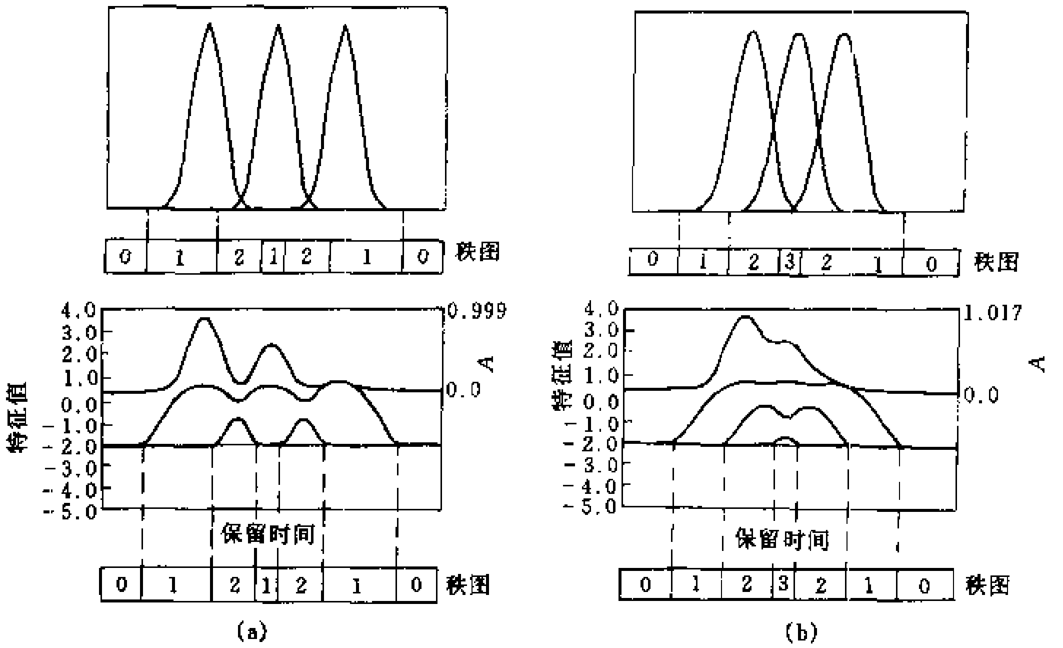


图 6-46　借特征结构跟踪分析技术所得的两个 3 组分分析体系在色谱保留时间轴上的秩图

(a) 两两重叠的色谱峰，局部秩最高为 2；(b) 3 组分重叠色谱峰局部秩最高为 3

设所得到的二维色谱数据为 $\boldsymbol{X}$，如果还考虑到仪器的背景，似可以把此矩阵分成两部分，即

$$\boldsymbol{X}=\boldsymbol{X}_c+\boldsymbol{X}_b+\boldsymbol{E} \tag{6-316}$$

式中，$\boldsymbol{X}_c$ 表示二维色谱数据的主要信息，它们来自体系存在的化学物质，$\boldsymbol{E}$ 为随机误差，而 $\boldsymbol{X}_b$ 则表示来自仪器的背景，因在考虑光谱背景的同时还需考虑色谱方向的漂移，所以它可由以下式表出，

$$\boldsymbol{X}_b=\boldsymbol{t}\boldsymbol{1}^t+\boldsymbol{1}\boldsymbol{s}^t \tag{6-317}$$

在此我们将仪器的色谱漂移和光谱背景分别用 $\boldsymbol{t}$ 和 $\boldsymbol{s}^t$ 来表示，式中的 $\boldsymbol{1}^t$ 和 $\boldsymbol{1}$ 为所有元素皆为 1 的矢量，其维数由二维色谱数据的列数和行数来决定。从式（6-317）可以看出，$\boldsymbol{X}_b$ 为一秩为二的矩阵。

为讨论方便，将在以下的叙述中将随机误差忽略，于是有

$$\boldsymbol{X}=\boldsymbol{X}_c+\boldsymbol{t}\boldsymbol{1}^t+\boldsymbol{1}\boldsymbol{s}^t \tag{6-318}$$

即

$$X_{ij}=X_{c,ij}+t_i+s_j \tag{6-319}$$

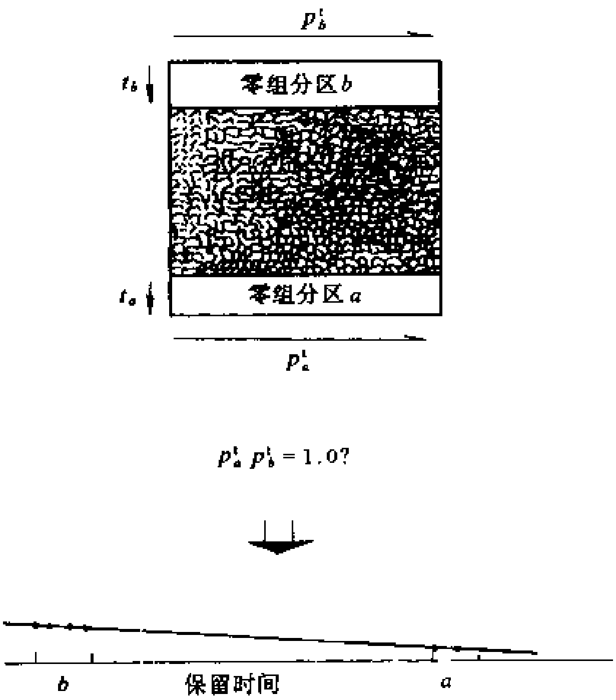


图 6-47 背景漂移扣除示意图

如果通过用特征投影图检出了在原始数据中存在背景信息，背景扣除就可以根据以下步骤来进行（参见图 6-47）。

① 采用局部主成分分析方法对色谱峰流出前和流出后的零组分区部分进行主成分分析，分别取出两个部分的第一个载荷矢量进行相合比较，即

$$p_b^t p_a \approx 1 \tag{6-320}$$

式中，p_b 和 p_a 分别表示色谱峰流出前和流出后的第一个载荷矢量，如果它们的相合系数接近于 1，这说明已找到了背景光谱，也就是说找到了式（6-318）中的 s^t。

② 采用上述主成分分析所得的两个部分的第一个得分矢量，或一有代表性的波长段，来对整段保留时间点进行最小二乘拟合，以得到式（6-318）中的 t。

③ 用如此得到的 s^t 和 t 来进行背景扣除，即

$$X_c = X - (t\mathbf{1}^t + \mathbf{1}s^t)$$

因为本文所述的方法是由直线最小二乘求出的，所以，对于一个较复杂的二维色谱数据，需要采用分段扣除的办法，这一情况可由图 6-48 表出。采用此法对几个不同的二维色谱体系进行了背景漂移的扣除，皆得到较好结果，其中最有代表性的是对叶绿素 a 降解体系的解析，读者可参阅文献[168]的图 9 和图 10。在背景扣除的方法中，经常有人提到双中心化方法，在文献[168]中对双中心化方法也进行了较详细的讨论，发现该法的确可以消除二维色谱数据的背景的影响，但是，它却引起 X_c 各局部区域的秩发生变化，尽管它可保持整个矩阵的秩不发生变化。此外，它还将破坏二维色谱数据的非负性，使经双中心化校正后的矩阵出现负元素，而不利于下一步的分辨解析。

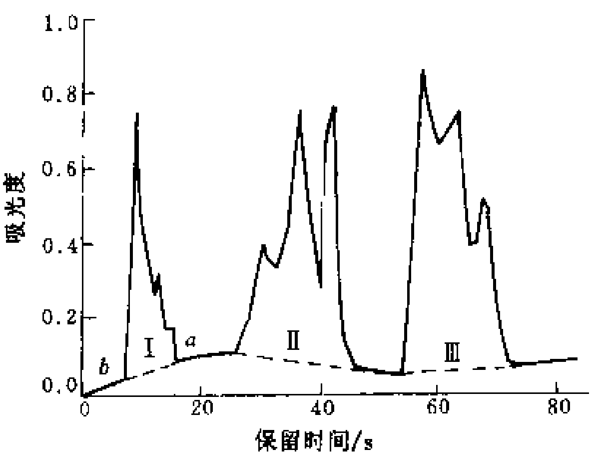


图 6-48 分段背影漂移扣除方法示意图

此法只讨论了色谱方向漂移为直线的情况，因此，可以说只能适用于一些漂移不严重的二维色谱数据的背景扣除，对于某些色谱漂移很严重的体系，色谱方向漂移可能不是直线，而且，光谱背景也可能因为色谱的梯度洗脱等原因而发生改变，对于这样的体系，如何进行有效的背景扣除仍为一值得研究的重要课题。希望能引起广大同行的关注。

4．选择性信息与满秩分辨

正如前面所讨论的，选择性信息对于二维色谱数据具有重要意义，它可使一些常见的重叠模式得到唯一解，这就是以下将要讨论的满秩分辨方法。

从前面的讨论可知，借特征结构跟踪分析（ETA）和特征投影图可以找到选择性信息，即单组分的纯组分光谱或色谱区域，这些选择性信息找出对二维色谱数据的分辨将带来很有用的信息。在讨论目标转换迭代因子分析法（ITTFA）时曾指出，如果找到了一个满秩旋转矩阵 $\boldsymbol{R}$，也就可将主成分分析所得到的抽象光谱和抽象色谱转换成为体系的纯光谱和纯色谱，所以，很多二维色谱数据的分辨方法的目的也就在于找到这个转换矩阵 $\boldsymbol{R}$。下面将介绍，HELP 方法的满秩分辨是怎样利用选择性信息来唯一求得这一转换矩阵 $\boldsymbol{R}$ 的，对于各组分浓度分布矢量的情况，有

$$\boldsymbol{c}_i=\boldsymbol{T}\boldsymbol{r}_i \tag{6-321}$$

从式（6-321）可以看出，因为 $\boldsymbol{T}$ 矩阵可由主成分分析求出，如果已知道 $\boldsymbol{c}_i$（$i=1$，…，n）的某些已知信息，就可以利用这些信息来求得 $\boldsymbol{r}_i$，求得 $\boldsymbol{r}_i$ 以后，再利用此式算出整个 $\boldsymbol{c}_i$ 来。问题的关键是怎样来利用所谓的"某些已知信息"？

从前面的讨论可知，利用演进特征投影图和特征结构跟踪分析方法，可以得到某些组分的选择性信息和秩图，对于图 6-49 所示出的三组分体系，可以很容易地找到第一流出组分和第三流出组分的选择性区域和它们的零浓度区域（参见图 6-49 的阴影部分），现以第一流出组分为例来说明 HELP 方法的满秩分辨过程。首先找到它的选择性区域，记为 $\boldsymbol{c}_{s,1}$，然后找到它的零浓度区域，记为 $\boldsymbol{c}_{0,1}$。因为在第一流出组分的选择性区域里，只有它一个组分存在，所以，此时的色谱就是该组分的纯色谱浓度分布，此外，在该组分的零浓度区域中，很明显，$\boldsymbol{c}_{0,1}$的每一个元素都等于零，把这两部分合起来（参见图 6-49 的下面部分），并把它记为 $\boldsymbol{c}_{0+s,1}$，这样就找到了第一流出组分的"某些已知信息"。于是可把式（6-321）改写成

$$\boldsymbol{c}_{0+s,1}=\boldsymbol{T}_{0+s}\boldsymbol{r}_1 \tag{6-322}$$

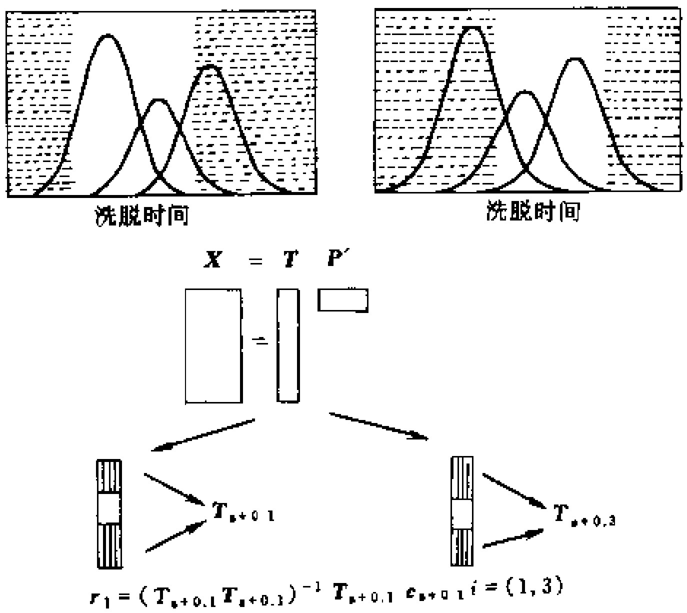


图 6-49 满秩分辨过程示意图

在此，$\boldsymbol{T}_{0+s}$表示得分矩阵 $\boldsymbol{T}$ 中的与第一流出组分的选择性区域和零浓度区域的相对应部分（参见图 6-49），注意到选择性区域包含了第一流出组分的信息，而零浓度区域又包含了其余两个组分的信息，所以，这样得到的 $\boldsymbol{T}_{0+s}$虽说只是 $\boldsymbol{T}$ 矩阵的一个子矩阵，但它实际上已包含了 3 个组分的信息，为一满秩矩阵，故可求得唯一解。当然，由于只为 $\boldsymbol{T}$ 矩阵一部分，它就不能保持 $\boldsymbol{T}$ 矩阵的正交性了。因为 $\boldsymbol{T}_{0+s}$为一满秩矩阵，而 $\boldsymbol{c}_{0+s,1}$又是已知的，这样就可以很容易地利

用最小二乘方法来对式（6-322）求解了，即

$$\boldsymbol{r}_1=(\boldsymbol{T}^{t}_{0-s}\boldsymbol{T}_{0+s})\boldsymbol{T}^{t}_{0-s}\boldsymbol{c}_{0\cdots1} \tag{6-323}$$

求得 $\boldsymbol{r}_1$ 后，再利用式 $\boldsymbol{c}_1=\boldsymbol{T}\boldsymbol{r}_1$ 即可求得整个 $\boldsymbol{c}_1$ 来。对于 $\boldsymbol{c}_3$ 可采用同样的方法，先找到它的选择性区域和零浓度区域以得到相应的 $\boldsymbol{T}$ 矩阵子矩阵，同样采用 $\boldsymbol{c}_3=\boldsymbol{T}\boldsymbol{r}_3$ 式即可求得 $\boldsymbol{c}_3$ 来。

求出 $\boldsymbol{c}_1$ 和 $\boldsymbol{c}_3$ 后，在选择性区域中可得到 $\boldsymbol{s}_1^{t}$ 和 $\boldsymbol{s}_3^{t}$，利用下式即可得到组分 2 的二维色谱矩阵，

$$\boldsymbol{X}_2=\boldsymbol{X}-(\boldsymbol{c}_1\boldsymbol{s}_1^{t}+\boldsymbol{c}_3\boldsymbol{s}_3^{t}) \tag{6-324}$$

因 $\boldsymbol{X}_2$ 为一单组分的二维色谱矩阵，为一双线性矩阵，对其直接分解即可求得它的纯组分光谱和色谱来。在此值得提出的是，上述满秩分解算法是可以推广到多于三组分的混合体系的。我们将此推广称为组分剥离过程，这一过程可由图 6-50 形象表示出，有关详细的讨论可参阅文献[170]。

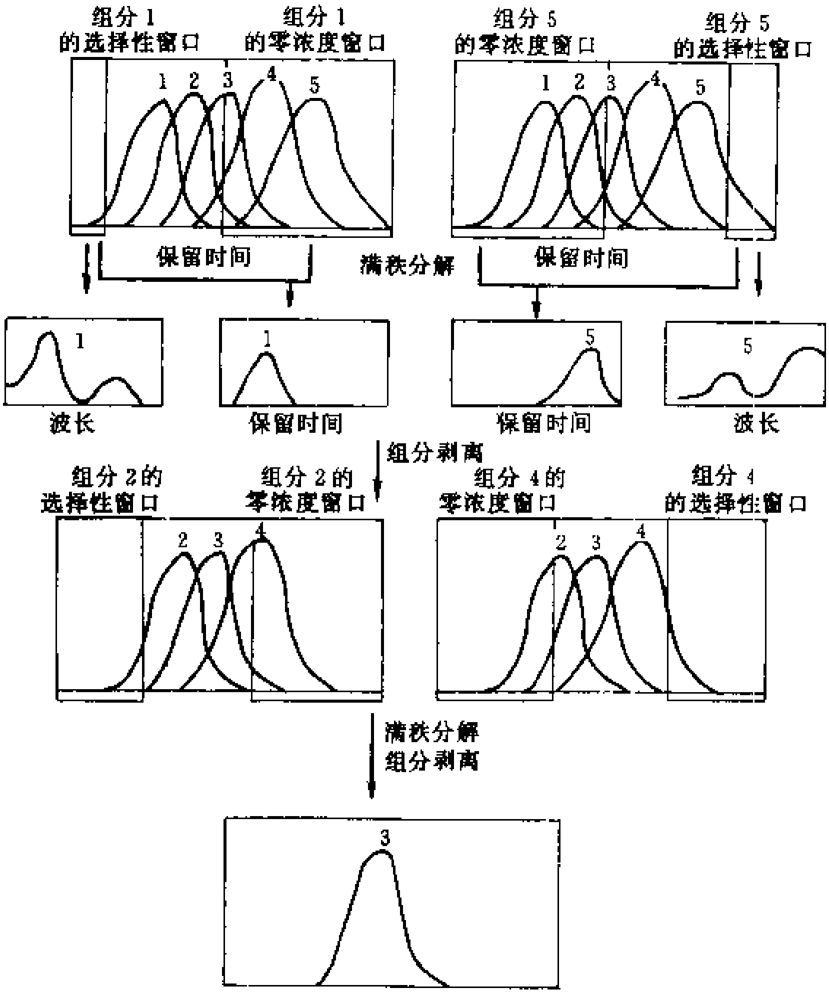


图 6-50　组分剥离过程示意图

首先对已知选择性信息和零浓度区的组分 1 和组分 5 进行满秩分辨，可得它们的纯光谱及纯色谱图，然后消除它们的影响，使该体系成为一个 3 组分体系，继续使用此法，可将此 5 组分体系全部分辨

【基本解析步骤】

综上所述，整个 HELP 算法将有以下几个步骤来完成：

① 背景的检测和扣除。利用演进特征投影图对要分辨的数据进行分析，如果零组分区的

光谱投影点没有集中于原点，则表示存在背景，然后利用零组分区的信息来扣除背景。

② 数据预处理与体系组分数的确定。对于存在有不等性方差噪声的数据，在用特征结构跟踪分析（ETA）来确定体系的秩图之前，有必要进行预处理。另外，在决定体系的组分数时，同样有必要利用演进特征投影图，实因此图对各组分的依次流出将给出很重要的信息，以帮助证实体系的组分数。

③ 选择性信息和零浓度区的确定。同时利用演进特征投影图和特征结构跟踪分析所得的秩图来确定选择性区域以及该组分的零浓度区域。

④ 纯组分光谱和色谱的满秩分辨。利用满秩分辨技术对已知选择性信息和零浓度区域的组分进行分辨，求得其转换矢量以获得该组分的纯光谱和纯色谱。对于组分数大于 3 的多组分体系，可采用组分剥离技术来逐步进行。

⑤ 对分辨所得结果进行证实。将分辨所得的各纯光谱与纯色谱乘积的线性加合结果与实际数据进行比较，以进一步确证所得结果的可靠性。

【讨论】

从以上所讨论的这五步可以看出，直观推导式演进特征投影法（HELP）是以逐步推导的方式，层层深入地来解析二维色谱数据，从背景的检出和扣除、不等性方差噪声的预处理、选择性信息的确定、秩图和色谱流出模式的确定一直到最后的满秩分辨，每一步都是从直观的特征投影图、局部因子分析和特征结构跟踪分析的所得的形象结果来进行的，这也是我们将此法称之为直观推导式方法的原因。

在此讨论的渐进因子分析法（EFA）、固定尺寸移动窗口渐进因子分析法（FSMWEFA）以及本节讨论的直观推导式演进特征投影法（HELP）与在前一节中讨论的用主成分分析法直接对整个矩阵进行体系组分数确定的方法（在本书中直接称之为“全局主成分分析法”）的最大不同之点就在于它们是着眼于局部的信息。由于色谱具有分离特性，所以，对于具有一定特性的联用色谱的二维数据，一般说来，这些渐进的、窗口的分析方法要明显优于那些一次性的、全局的分析方法，而使得一些具有特定的二维色谱数据可以得到唯一解，为黑色分析体系的精确解析提供了可能性。这是因为这些渐进的、窗口的分析方法很充分地利用了色谱的分离特性，将一个较复杂的多组分体系首先化解成若干个相对简单的子体系，而将一些影响分辨结果的重要因素，如光谱和色谱重叠性、组分之间的相对浓度比、噪声水平的确定等降低到尽可能的低水平。当然，对于一般的黑色分析体系的唯一精确解仍是化学计量学研究的一个热点问题，这在以后的讨论中将逐步展开。

由于直观推导式渐进特征投影法（HELP）具有从原始数据的背景扣除一直到色谱重叠峰的最后分辨的所有解析步骤，所以近年来得到了广泛地应用，它被应用于药物纯度的直接检验、光化学研究中的叶绿素 a 的降解分析[171]、红外光谱分辨解析[172]、复杂的环境分析中空气实际样本的多环芳烃的定性定量分析[173]等，取得了令人兴奋的成果，是一个值得重视的研究新领域，以下我们将以环境分析中空气实际样本的多环芳烃的定性定量分析为例来对 HELP 方法解析的全过程给出较详细的说明，以求读者对 HELP 方法有一个整体印象。

多环芳烃是一类广泛存在于人类环境中的有机污染物的统称。在一千多种能使动物致癌的物质中，多环芳烃占三分之一多，因此，对多环芳烃的研究在理论和实际上都有重要意义。我们用 HELP 方法对香港城市大气颗粒物中进行了多环芳烃化合物的定性定量分析，分析结果说明，在该样本中存在 47 种化学成分，其中有 18 种为多环芳烃化合物及其衍生物，分辨

结果与毛细管气相色谱与质谱联用仪所得结果一致。

图 6-51 分别示出了该空气样本在 254nm、260nm 及 310nm 上测得的色谱图。从图可以看出，每个图中都只有不足 20 个色谱峰。用 HPLC-DAD 所带的常规标准光谱库搜索发现，只有 4～5 个色谱峰位置的光谱与标准光谱匹配程度较高，这表明样本的实际组成可能很复杂，所得的色谱峰可能大多为未完全分离的重叠峰。用 HELP 方法对此样本的解析结果示于图 6-52，发现此样本确实含有 47 种化学成分，其中有 18 种为多环芳烃化合物及其衍生物（参见图 6-52 中由数字表出的色谱峰）。下面以保留时间分别为（20.47～24.53）min 和（12.00～16.00）min 的两段峰簇为例来说明样本的解析全过程［参见图 6-51（a）中标出部分］。

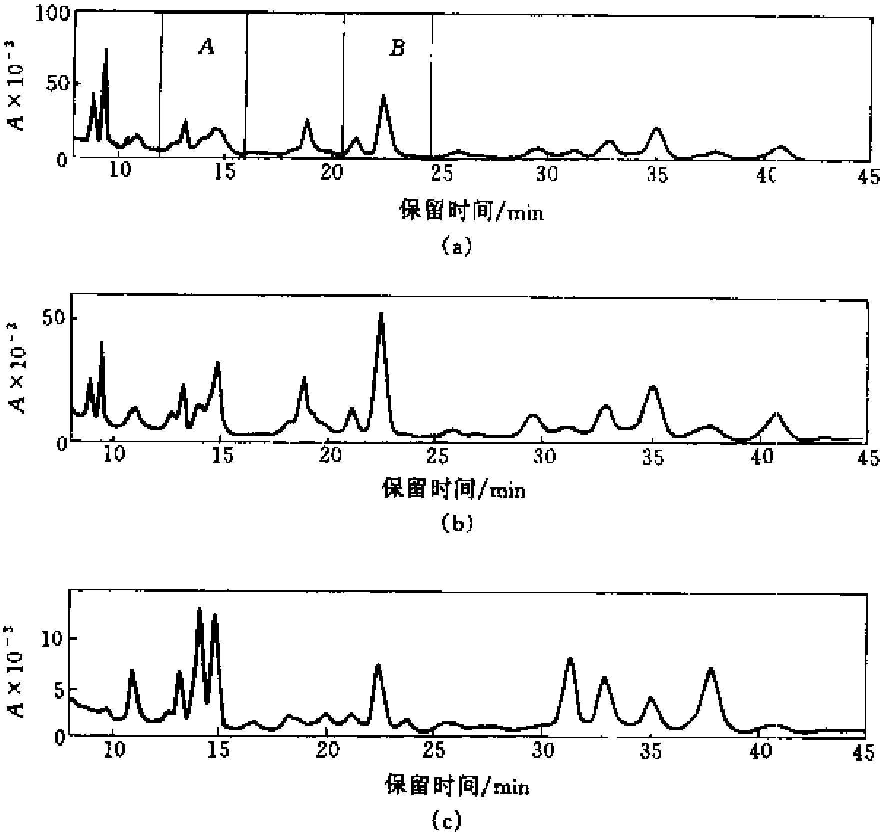


图 6-51　空气样本在波长为 254nm（a）、260nm（b）、310nm（c）的色谱流出曲线

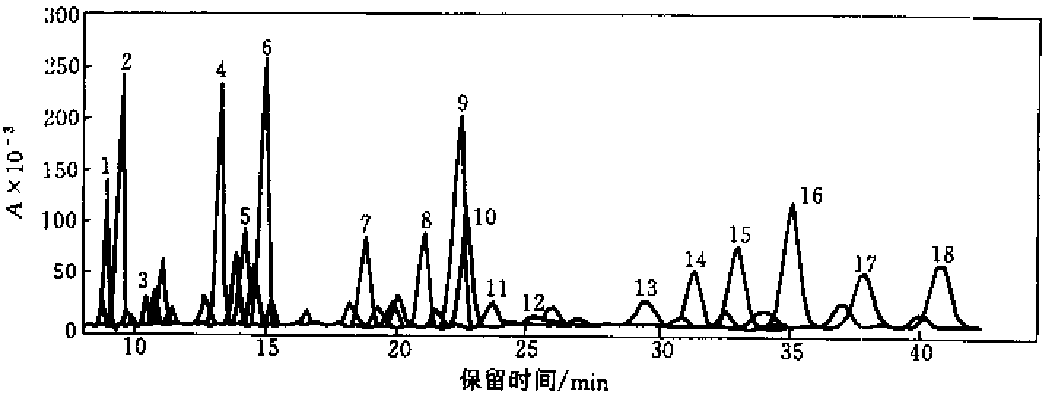


图 6-52　样本的解析结果

（图中数字表示已确认的 PAH）

1—菲；2—菲；3—蒽；4—荧蒽；5—芘；6—芘；7—苯并菲；8　环戊二烯并芘；9—䓛；10—苯并［a］蒽；11—芘；12—芘；13—苯并菲；14—苯并［j］荧蒽；15—苯并［e］芘；16—苯并［b］荧蒽；17—苯并［k］荧蒽；18—苯并［a］芘

因实际样本的量测一般都存在基线漂移，所以，在进行分辨以前，首先要扣除漂移背景。对保留时间(20.47～24.53)min 所示的峰簇前后的两个零组分区进行局部主成分分析所得结果示于图 6-53，所得的两个第一载荷矢量的相合系数为 0.9945，显然，体系存在背景漂移，须用前述方法进行背景扣除。以后对该样本的其他色谱峰都是采用此法来预先进行背景扣除而后再进行分辨的。

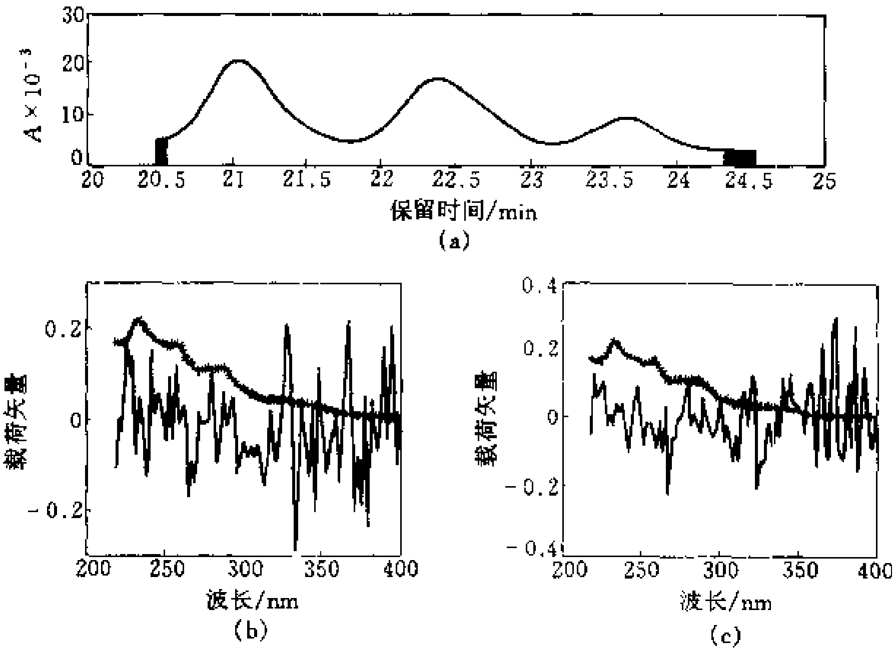


图 6-53　保留时间为 (20.47～21.53) min 的峰簇的背景扣除图

(a) 峰簇在 230nm 的色谱流出曲线（阴影部分表示零组分区）；(b)，(c) 对峰簇两端的零组分区进行 LPCA 所得的前两个载荷矢量，┴ 为第一个载荷矢量，— 为第二个载荷矢量

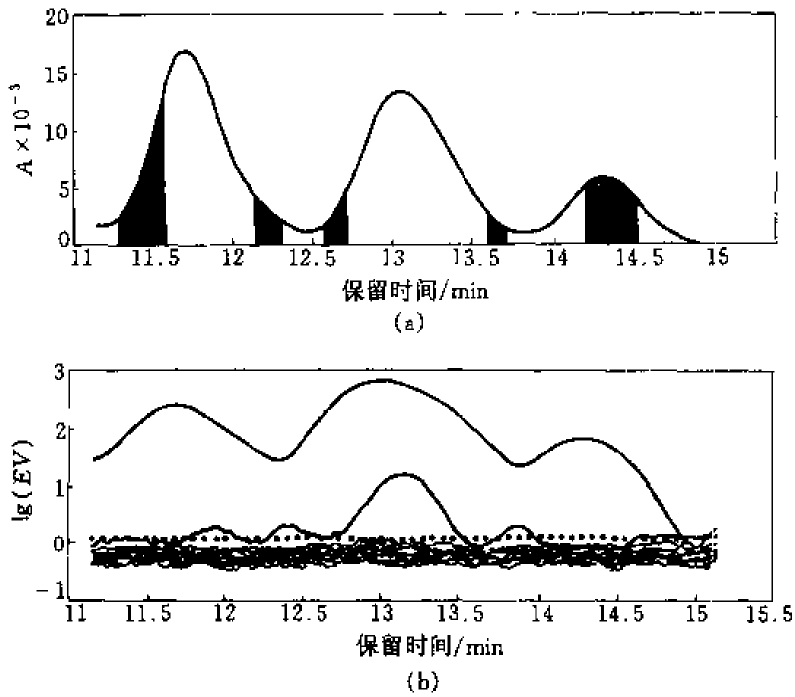


图 6-54　保留时间为 (20.47～24.53) min 的峰簇的解析

(a) 峰簇在 230 纳米的色谱流出曲线（图中阴影部分表示选择性区域）；

(b) 体系的特征结构跟踪分析图（…为噪声水平）

背景扣除后，进一步就要确定峰簇中组分数和选择性信息。建议采用特征投影图和特征

结构跟踪分析（ETA）两种技术共同来确定峰簇中的组分数和选择性信息，实因只有当两种技术所得结果相一致时，所得最终分辨结果才可靠。图 6-54 示出了保留时间（20.47～24.53）min 区域的峰簇特征结构跟踪分析所得的秩图（ETA 图）。从图 6-54 可以看出，此峰簇为 1 个五组分体系，且 5 个组分都存在选择性区域。从各个选择性区域中可容易地获得各组分的纯物种光谱，即对各选择性区域作局部主成分分析，继取其第一个载荷矢量。因为已经取得所有组分的纯物种光谱，然后直接用最小二乘方法即可得它们的相应纯组分色谱。图 6-55 示出了该峰簇的分辨结果，经与 PAH 的标准光谱比较可知，体系中的组分 1、3、4、5 分别为环戊二烯并苯、䓛、苯并［a］蒽及芘的衍生物，解析所得光谱与标准光谱吻合很好。

与上例相比，对保留时间（12.00～16.00）min 区域所包括的峰簇的定性分辨就较为困难。首先就是该峰簇的组分数难于确定。表 6-16 和图 6-56 分别示出了对该峰簇进行主成分分析所得的特征值和特征向量的结果。从表 6-16 看来，似乎该体系为 1 个 4 组分体系，实因从第

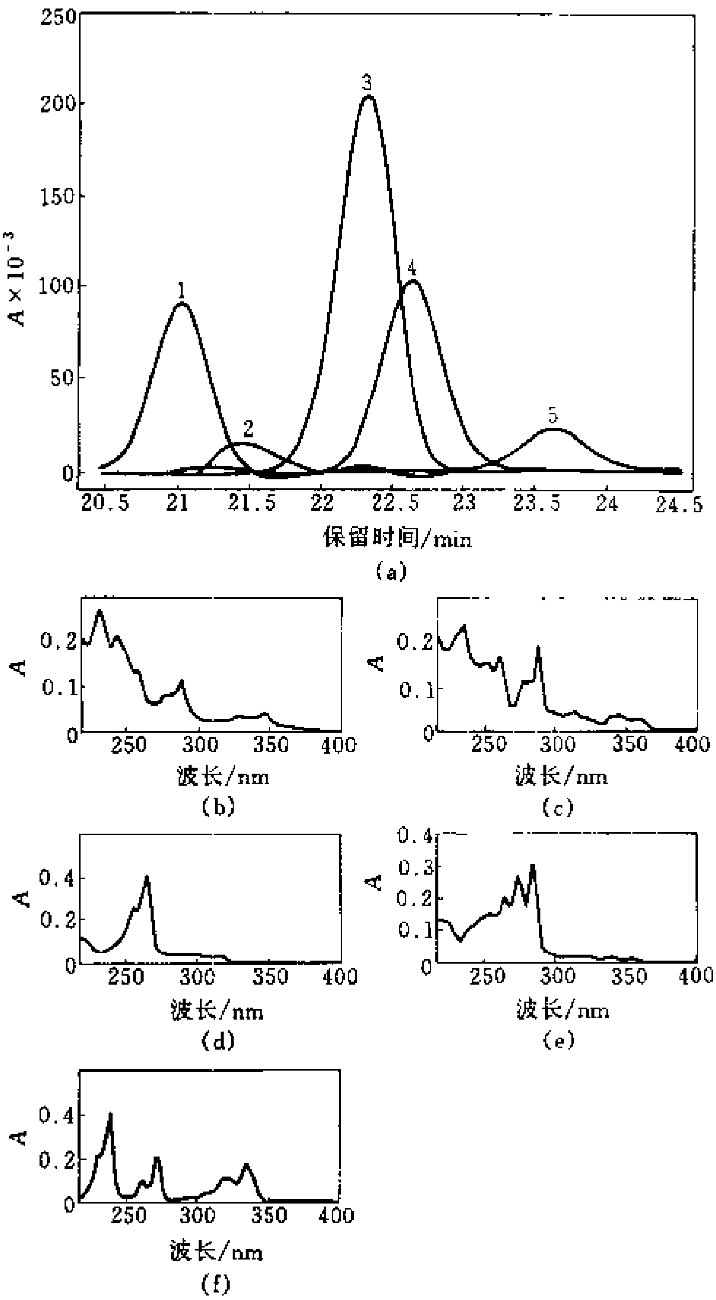


图 6-55　保留时间为（20.47～24.53）min 的峰簇的解析结果
(a) 解析所得的色谱流出曲线；(b) ～ (f) 解析所得 1～5 组分的光谱图

表 6-16 对（12.00～16.00）min 的峰簇进行 PCA 所得的特征值

序号	1	2	3	4	5	6	7	8	9
特征值	2.17	0.69	0.40	0.23	0.06	0.04	0.02	0.02	0.01

五个特征值以后变化就很小了；而从特征向量的结果（图 6-56）看来，似乎该体系又是 1 个 5 组分体系，实因第六个特征向量就很像噪声的形状了。可是，实际上该峰簇是 1 个七组分体系！图 6-57 给出了它的 ETA 图和特征投影图，两个图都指出该峰簇存在有五个选择性区域，即图 6-57（a）所示的 A、B、C、F 和 G 区，与其相对应，图 6-57（b）给出了它们在特征投影图的直线段［图中标出的 a、b、c、d 对应于图 6-57（a）中的 F，e 对应于图 6-57（a）中的 G］，而且这些直线段经延长都通过原点。注意到图 6-57（a）中的 D 和 E 所指示的两组分重叠区，说明该峰簇至少为一 7 组分体系，同时，从图 6-57（b）给出的演进特征投影图来分析，从 c 段到 d 段，投影曲线改变方向两次也证实了这一点。根据以上结果，可以初步确定该峰簇至少存在 7 个组分，有 ETA 图得到的各组分选择性信息和它们的零浓度区域，采用前述的满秩分辨技术，可先将这 5 个具有选择性信息的组分分辨出来，然后采用组分剥离技术将它们在此峰簇的贡献减去，继对剩余部分进行特征结构跟踪分析。对剩余部分进行特征结构跟踪分析的结果示于图 6-58，从图 6-58 可以看出，所剩部分确为一 2 组分体系。该峰簇的总的分辨结果示于图 6-59。在此峰簇的 7 个组分中，第二、三组分分别为荧蒽和芘，第六组分的纯光谱与芘的纯光谱很相似，可认为是芘的衍生物。值得指出的是，为验证上述所得结

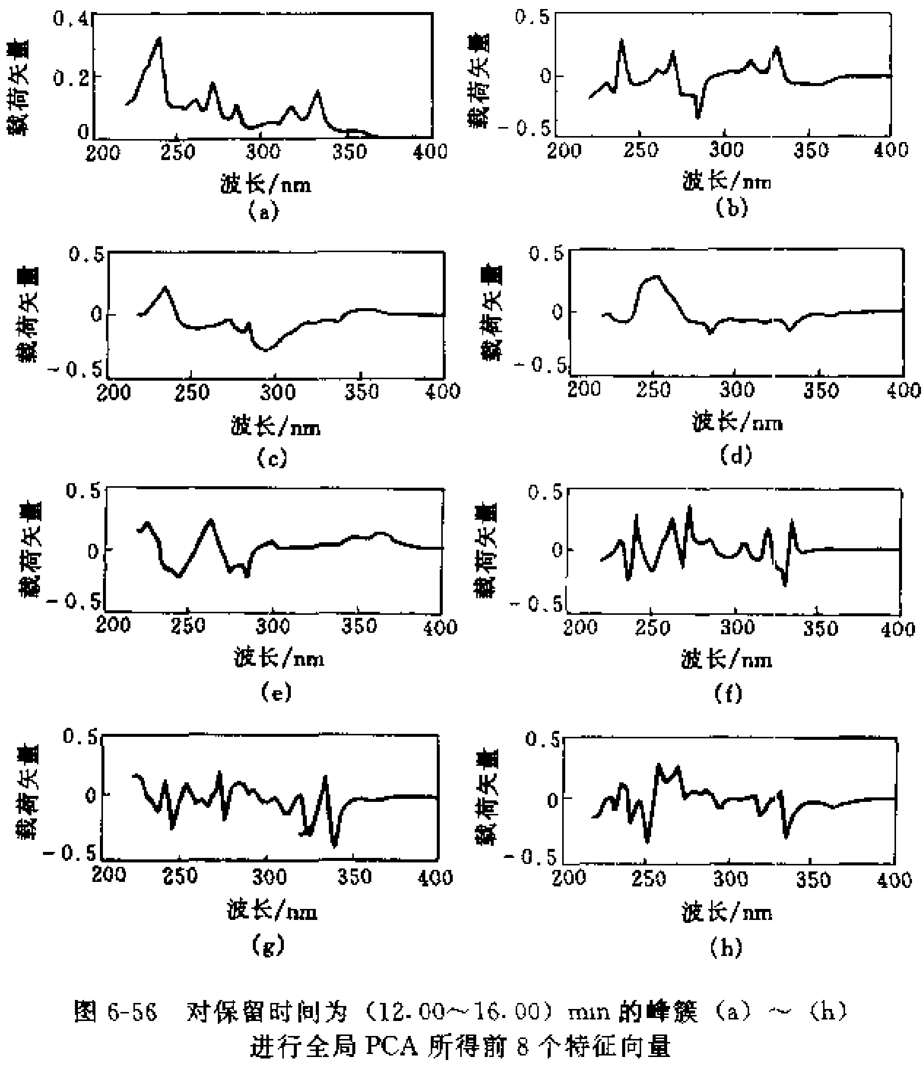
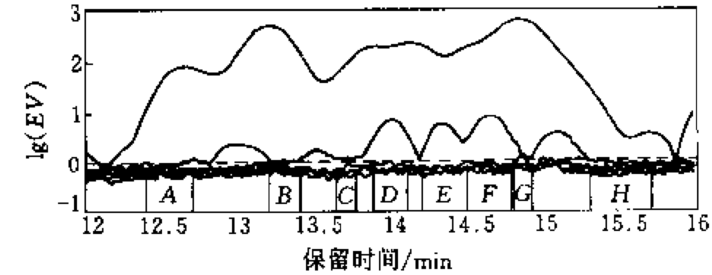
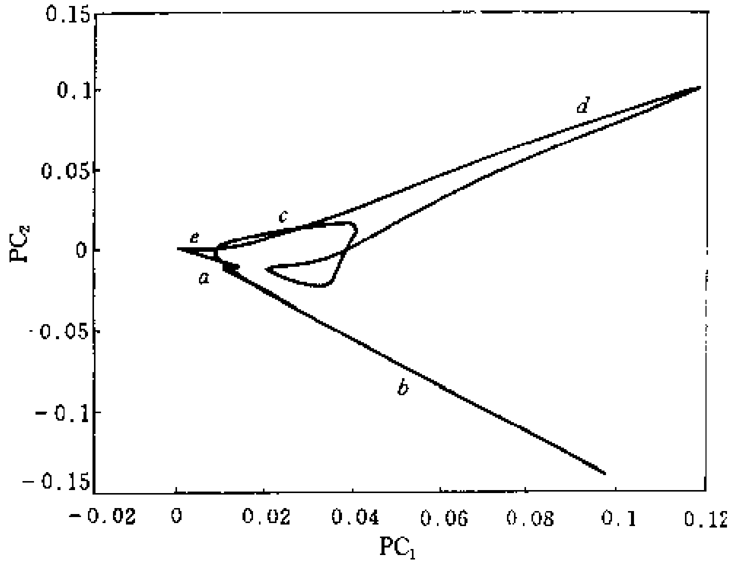


图 6-56 对保留时间为（12.00～16.00）min 的峰簇（a）～（h）进行全局 PCA 所得前 8 个特征向量

果，我们还做了毛细管色谱质谱联用分析，从其总离子流色谱中可得该栏本有40几个峰，对PAH的定性分析结果也与用HELP方法解析所得结果相一致，从而进一步说明了HELP方法的可靠性。



(a)



(b)

图 6-57 HELP方法确定保留时间为（12.00～16.00）min的峰簇的组分数

(a) 体系的ETA图（……为噪声水平）；(b) 体系的特征投影图

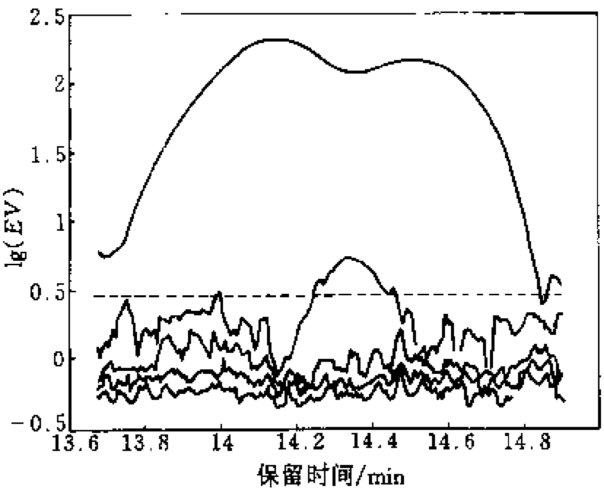


图 6-58 通过满秩分辨消去图 6-57 中具有选择性区域的5个组分后峰簇剩余部分的ETA图

----为噪声水平

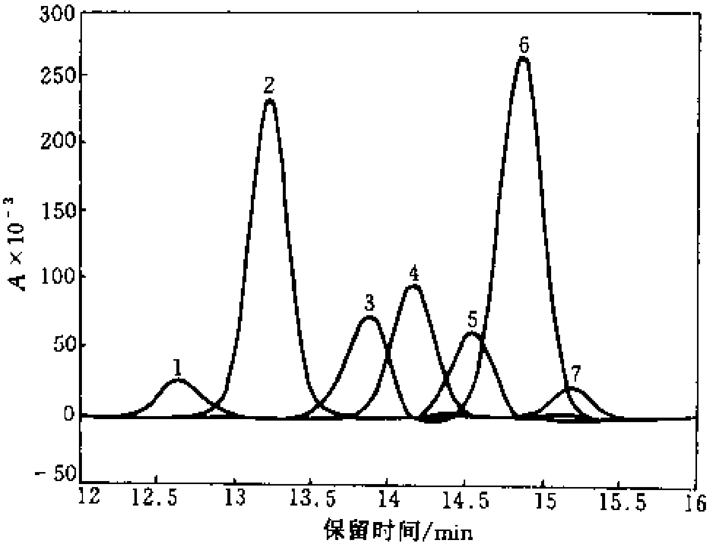
未知复杂样本如生物样本、天然药物、环境样本等的分析在生物化学、药物化学、环境化学等领域有着极其重要的地位，对这些样本的直接快速的定性定量分析可以说是当前分析化学的热点和难点，从上例看来，化学计量学近年来发展的黑色分析体系的解析方法，其中特别是HELP方法为这些未知复杂样本的直接快速分析提供了一个崭新的途径，是一类极有发展前途的分析方法。

（六）正交投影分辨法

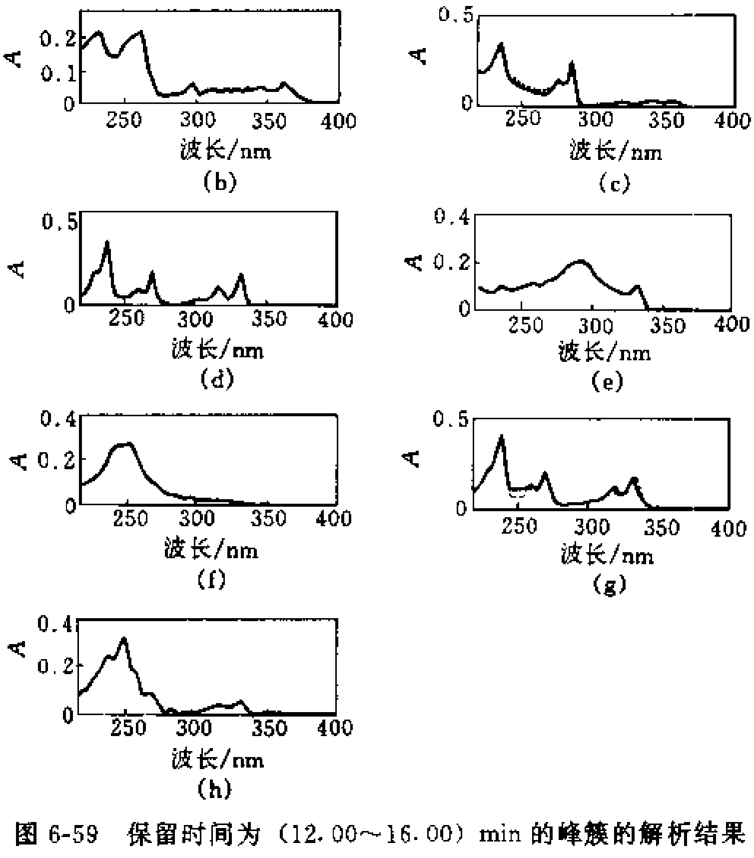
【基本思路】

正交投影分辨算法（OPR）是基于Lorber提出的正交投影[174]的基本思路而提出，在结构上与Malinowski提出的窗口因子分析法

(WFA）类似，它也是利用已知的体系中含 $n-1$ 个物种的窗口来对第 n 个物种进行正交投影以求得其色谱流出信息。最简单的例子是在小色谱峰完全被大色谱峰覆盖的情况中，利用已经知道的大色谱峰相对应的纯物种光谱来求得小色谱峰的形状[175]。实际上，这一算法可容易地拓广至一般色谱重叠情况的分辨，对于最常见的三组分体系分辨的基本思路图示于图 6-60，在以后我们还可以看到，正交投影技术还可用于来诊断不同的色谱重叠模式，从这一角度来说，它明显优于 EFA[176]。下面我们将对正交投影分辨算法给出介绍。



(a)



(b) (c) (d) (e) (f) (g) (h)

图 6-59 保留时间为（12.00～16.00）min 的峰簇的解析结果

(a) 色谱流出曲线；(b) ～ (h) 各组分的光谱（——解析所得光谱；……标准光谱）

【基本算法】

正交投影分辨算法由以下几个步骤构成：

① 利用秩图来定义体系中不同的含（$n-1$）个物种的窗口（参见图 6-60）。

② 对已定义的（$n-1$）个物种的窗口所对应的子矩阵进行主成分分解以得到（$n-1$）个标准化的且相互正交的载荷矢量（或称为抽象光谱）。

③ 构成由这（$n-1$）个载荷矢量组成的投影矩阵 $\boldsymbol{M}_k$

$$\boldsymbol{M}_k=(\boldsymbol{I}-\boldsymbol{P}_k^{(n-1)}\boldsymbol{P}_k^{(n-1)\text{t}}) \tag{6-325}$$

式中，$\boldsymbol{M}_k(k=1,2,\cdots,n)$为对第 k 个组分求其浓度分布（或色谱）的投影矩阵；$\boldsymbol{P}_k^{(n-1)}=[\boldsymbol{p}_{1k},\boldsymbol{p}_{2k},\cdots,\boldsymbol{p}_{(n-1)k}]$则表示由对已定义的不包括第 k 物种的（$n-1$）个物种的窗口所对应的子矩阵进行主成分分解而得到的（$n-1$）个标准正交的载荷矢量；$\boldsymbol{I}$ 为一相应大小的单位矩阵。

④ 对于不同的组分，用不同的投影矩阵 $\boldsymbol{M}_k(k=1,2,\cdots,n)$对混合物量测矩阵 $\boldsymbol{X}$ 的每一条光谱矢量 $\boldsymbol{x}_j^{\text{t}}$ 进行投影运算，

$$\begin{aligned}\boldsymbol{x}_j^{\text{t}*}&=\boldsymbol{x}_j^{\text{t}}\boldsymbol{M}_k=(c_1\boldsymbol{s}_1+\cdots+c_k\boldsymbol{s}_k+\cdots c_n\boldsymbol{s}_n)_j^{\text{t}}(\boldsymbol{I}-\boldsymbol{P}_k^{(n-1)}\boldsymbol{P}_k^{(n-1)\text{t}})\\&=(c_k\boldsymbol{s}_k)_j^{\text{t}}\boldsymbol{M}_k=(c_k\boldsymbol{s}_k^*)_j^{\text{t}}\end{aligned} \tag{6-326}$$

式中，下标 j 表示混合物量测矩阵 $\boldsymbol{X}$ 的第 j 条光谱，关于(6-55)式的证明在本节的最后。

⑤ 因为光谱矢量 $\boldsymbol{x}_j^{\text{t}}$ 经投影运算之后得到的 $\boldsymbol{x}_j^{\text{t}*}$ 是一个只剩下 c_k 和 $\boldsymbol{s}_k^*$ 信息，所以，如果对式(6-326)两边同时进行取模运算，立即可得

$$\|\boldsymbol{x}_j^{\text{t}*}\|=\|(c_k\boldsymbol{s}_k^*)_j^{\text{t}}\| \tag{6-327}$$

即

$$\|\boldsymbol{x}_j^{\text{t}*}\|=(c_k)_j\|(\boldsymbol{s}_k^*)^{\text{t}}\| \qquad (j=1,\cdots,m) \tag{6-328}$$

式中，j 为光谱的波长点数。上式左边的 $\|\boldsymbol{x}_j^{\text{t}*}\|$ 为已知，而右边的$(c_k)_j\|(\boldsymbol{s}_k^*)^{\text{t}}\|$则是我们要求的。在此还特别值得提出的是，对于 $\|(\boldsymbol{s}_k^*)(c_k)_j\|$ 中的 $\|(\boldsymbol{s}_k^*)_j^{\text{t}}\|$ 我们略去了下标 j，这是因为对于不同的光谱矢量 $\boldsymbol{x}_j^{\text{t}}$ 中的 $\boldsymbol{s}_k$ 是同一的，故 $\boldsymbol{s}_k^*$ 也是同一的。

如果将 m 个数 $\|\boldsymbol{x}_j^{\text{t}*}\|(j=1,\cdots,m)$收集起来成为一个矢量，则得到矢量$(\|\boldsymbol{x}_1^{\text{t}*}\|,\cdots,\|\boldsymbol{x}_m^{\text{t}*}\|)$，再根据式(6-326)得

$$\begin{aligned}&(\|\boldsymbol{x}_1^{\text{t}*}\|,\cdots,\|\boldsymbol{x}_m^{\text{t}*}\|)\\&=((c_k)_1\|(\boldsymbol{s}_k^*)_1^{\text{t}}\|,\cdots,(c_k)_m\|(\boldsymbol{s}_k^*)_m^{\text{t}}\|)\\&=\|(\boldsymbol{s}_k^*)_1^{\text{t}}\|((c_k)_1,\cdots,(c_k)_m)\end{aligned} \tag{6-329}$$

即要求的第 k 组分的浓度分布乘上一个常数 $\|(\boldsymbol{s}_k^*)_1^{\text{t}}\|$，因为 $\boldsymbol{s}_k$ 是未知的，所以这个常数 $\|(\boldsymbol{s}_k^*)_1^{\text{t}}\|$ 亦为未知。因此，式(6-329)得到的浓度分布并不是实际的第 k 组分的浓度分布，而是它与一未知常数的乘积。

⑥ 重复第④步和第⑤步，求得每个物种的浓度分布模式，即可用最小二乘技术算出纯物种的光谱来，即

$$\boldsymbol{S}^{\text{t}}=(\boldsymbol{C}^{\text{t}}\boldsymbol{C})^{-1}\boldsymbol{C}^{\text{t}}\boldsymbol{X} \tag{6-330}$$

在此 $\boldsymbol{C}$ 矩阵虽不是实际的浓度分布，但并不影响纯光谱矩阵 $\boldsymbol{S}^{\text{t}}$ 的求得，实因样求得的 $\boldsymbol{S}^{\text{t}}$ 只是在大小上与实际的 $\boldsymbol{S}^{\text{t}}$ 不同，只需将它们进行标准化处理就行了。

以上讨论的用正交投影方法来进行黑色分析体系解析的过程基本与 Malinowski 提出的窗口因子分析法类似，整个过程可由图 6-60 形象地表示出。

【讨论】

式(6-55)的证明（在此只给出 k 等于 n 时的情况，其余可类推）：

设 $\boldsymbol{X}^{(n-1)}$为一只含（$n-1$）个物种的子矩阵，$\boldsymbol{X}^{(n-1)+}$ 是它的广义逆，根据广义逆的性质（参

见第十一章)，有

$$X^{(n-1)}X^{(n-1)-}X^{(n-1)}=X^{(n-1)} \tag{6-331}$$

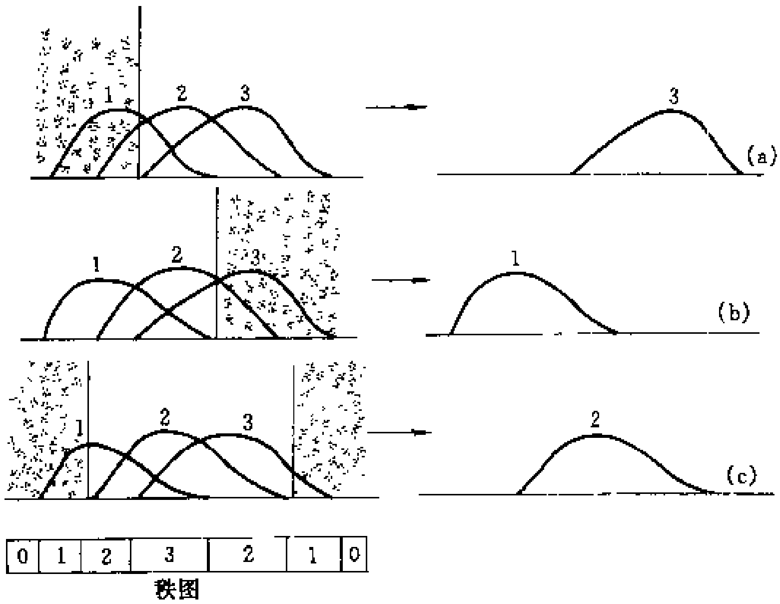


图 6-60 正交投影算法分辨常见三组分重叠体系的示意图

首先采用特征结构跟踪分析法(ETA)得到体系的化学秩图；继找到 1 个二组分区域以构成投影矩阵，以得到相应未包括在内组分的纯色谱

(a) 以第一、二组分信息构成正交投影阵，求得第三组分之纯色谱；

(b) 以第二、三组分构成投影阵求第一组分的纯色谱；

(c) 以第一、三组分构成投影阵，求第二组分的纯色谱

如果写成矢量形式，则有

$$x_i^{(n-1)t}X^{(n-1)+}X^{(n-1)}=x_i^{(n-1)t} \tag{6-332}$$

式中，$x_i^{(n-1)t}$为矩阵 $X^{(n-1)}$中的第 i 行。

据此可以构造一个正交投影矩阵

$$M=(I-X^{(n-1)-}X^{(n-1)}) \tag{6-333}$$

式(6-333)所示的正交投影矩阵实际上就是前述所示的正交投影矩阵、对矩阵 $X^{(n-1)}$进行主成分分解，可得

$$X^{(n-1)}=U^{(n-1)}G^{(n-1)}P^{(n-1)t} \tag{6-334}$$

在此 U 为一列正交矩阵，即所谓得分矩阵；G 为一对角矩阵，收集了 X 矩阵的非噪声特征值，如与本章的第一节所说的特征值对应的话，是 X 矩阵的协方差阵所得特征值的平方根；P^t 为一行正交矩阵，即所谓载荷矩阵，对式(6-334)求广义逆，则可得

$$\begin{aligned}(X^{(n-1)})^{+}&=(U^{(n-1)}G^{(n-1)}P^{(n-1)t})^{-1}\\&=(P^{(n-1)})(G^{(n-1)})^{-1}U^{(n-1)t}\end{aligned} \tag{6-335}$$

亦即

$$\begin{aligned}(X^{(n-1)})^{+}X^{(n-1)}&=P^{(n-1)}(G^{(n-1)})^{-1}U^{(n-1)t}U^{(n-1)}G^{(n-1)}P^{(n-1)t}\\&=P^{(n-1)}P^{(n-1)t}\end{aligned} \tag{6-336}$$

$x_i^{(n-1)t}$ 可表示为$(n-1)$个纯组分光谱 $s_i^t(i=1,2,\cdots,n-1)$的线性组合。即

$$x_j^{(n-1)t}=(c_1s_1+\cdots+\cdots c_{n-1}s_{n-1})_j^t \tag{6-337}$$

而对于一个含 n 个物种的量测矩阵 X，它的每一行 x_i^t 可表为 n 个纯组分光谱 $s_i^t(i=1,2,\cdots,n)$

的线性组合，即

$$\begin{aligned} \boldsymbol{x}_i^t &= (c_1\boldsymbol{s}_1 + \cdots + c_{n-1}\boldsymbol{s}_{n-1} + c_n\boldsymbol{s}_n)_j^t \\ &= (\boldsymbol{x}_i^{(n-1)} + c_n\boldsymbol{s}_n)_j^t \end{aligned} \tag{6-338}$$

所以，式(6-326)可表示为

$$\begin{aligned} \boldsymbol{x}_j^{t*} &= \boldsymbol{x}_j^t\boldsymbol{M} = (c_1\boldsymbol{s}_1 + \cdots + c_{n-1}\boldsymbol{s}_{n-1} + c_n\boldsymbol{s}_n)_j^t(\boldsymbol{I} - \boldsymbol{P}^{(n-1)}\boldsymbol{P}^{(n-1)t}) \\ &= (\boldsymbol{x}_i^{(n-1)} + c_n\boldsymbol{s}_n)_j^t(\boldsymbol{I} - \boldsymbol{P}^{(n-1)}\boldsymbol{P}^{(n-1)t}) \\ &= \boldsymbol{x}_i^{(n-1)t} - \boldsymbol{x}_i^{(n-1)t}\boldsymbol{P}^{(n-1)}\boldsymbol{P}^{(n-1)t} + (c_n\boldsymbol{s}_n)_j^t(\boldsymbol{I} - \boldsymbol{P}^{(n-1)}\boldsymbol{P}^{(n-1)t}) \\ &= \boldsymbol{x}_i^{(n-1)t} - \boldsymbol{x}_i^{(n-1)t}\boldsymbol{X}^{(n-1)} + \boldsymbol{X}^{(n-1)} + (c_n\boldsymbol{s}_n)_j^t(\boldsymbol{I} - \boldsymbol{P}^{(n-1)}\boldsymbol{P}^{(n-1)t}) \\ &= \boldsymbol{x}_i^{(n-1)t} - \boldsymbol{x}_i^{(i-1)t} + (c_n\boldsymbol{s}_n)_j^t(\boldsymbol{I} - \boldsymbol{P}^{(n-1)}\boldsymbol{P}^{(n-1)t}) \\ &= (c_n\boldsymbol{s}_n)_j^t\boldsymbol{M} - (c_n\boldsymbol{s}_n^*)_j^t \end{aligned}$$

在此只需把下标 k 改为 n 就变成(6-326)式。

实际上我们还可以证明本节介绍正交投影分辨法与 Malinowski 提出的窗口因子分析法是同一的，参阅文献[177]。

（七）子窗口因子分析法

【基本思路】

前述分辨方法，如渐近因子分析法、窗口因子分析法及正交投影分辨法，都是以分辨纯组分色谱为其目标，而在本节介绍的子窗口因子分析法却是以分辨纯组分光谱为其目标。该法由 Manne 等提出[178]，其主要思路是利用在色谱方向上不同子窗口所含有一个且只含有一个相同纯物质光谱的特性，从而将该纯物质光谱分辨出来。这样的子窗口可由图 6-61 简要示出。

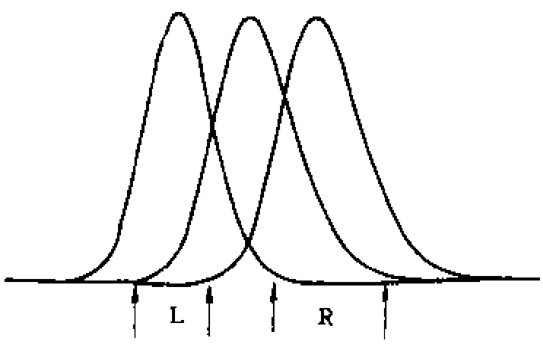


图 6-61　色谱方向的子窗口示意图
共同组分为中间的流出的组分；
L—中间组分的左窗口；
R　中间组分的右窗口

【基本算法】

子窗口因子分析算法可由以下几个步骤构成：

① 利用秩图来定义体系中不同的含有一个且只含有一个相同纯物质的两个不同子窗口（参见图 6-61）。

② 对已定义的两个子窗口所对应的子矩阵分别进行主成分分解，以得到两组标准正交基 $\boldsymbol{E}=\{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_m\}$和 $\boldsymbol{D}=\{\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n\}$，其中 m 和 n 分别表示两个子窗口所对应的子矩阵的组分数。设该同时含于两个子矩阵的纯光谱为 $\boldsymbol{s}$，因可被两组正交基同时线性表示，则有

$$\begin{aligned} \boldsymbol{s} &= \sum_{i=1}^{m}\alpha_i\boldsymbol{e}_i = \sum_{j=1}^{n}\beta_j\boldsymbol{d}_j \\ &= \boldsymbol{E}\alpha = \boldsymbol{D}\beta \end{aligned}$$

式中，$\alpha_i(i=1,2,\cdots m)$，$\beta_j(j=1,2,\cdots n)$，分别为线性组合系数。而且，有

$$\alpha = \boldsymbol{E}^t\boldsymbol{s} \qquad \beta = \boldsymbol{D}^t\boldsymbol{s}$$

③ 构一个目标函数 N，由 $\boldsymbol{E}^t\boldsymbol{E}=\boldsymbol{I}_m$；$\boldsymbol{D}^t\boldsymbol{D}=\boldsymbol{I}_n$，有

$$\begin{aligned} N = \|\boldsymbol{E}\alpha - \boldsymbol{D}\beta\|^2 &= \alpha^t\boldsymbol{E}^t\boldsymbol{E}\alpha + \beta^t\boldsymbol{D}^t\boldsymbol{D}\beta - 2\alpha^t\boldsymbol{E}^t\boldsymbol{D}\beta \\ &= 2 - 2\alpha^t\boldsymbol{E}^t\boldsymbol{D}\beta \\ &= 2 - 2f_1 \end{aligned}$$

式中，$f_1=\alpha^t E^t D\beta$。

④ 对上述目标函数求极小，实际上相当于对 $f_1=\alpha^t E^t D\beta$ 求极大。因为

$$\alpha=E^t s=E^t D\beta=E^t DD^t s-E^t DD^t E\alpha$$

$$\beta=D^t s=D^t E\alpha=D^t EE^t s=D^t EE^t D\beta$$

所以，α、β 分别为 E^tDD^tE 和 D^tEE^tD 的特征值为 1 的特征向量(参见第十一章)。同时

$$g=E\alpha=D\beta$$

即无论是求得 α 或 β，就可算出该同时含于两个子矩阵的纯光谱 s 来。

⑤ 只要没有含大峰包小峰的情况，就可为任一组分找到相应的两个子窗口，而依次得到其纯光谱 $s_i(i=1,2,\cdots,A)$，在此 A 为体系组分数。求得所有的纯组分的光谱即用式(6-339)可求得各组分的纯色谱来

$$C=XS(S^tS)^{-1} \tag{6-339}$$

【讨论】

因本法与前述分辨方法不同的地方在于它不是求纯组分的色谱而以求组分物质的纯光谱为目标，所以，可为联用色谱的二维数据的解析带来新内容，将此法与前述的正交投影分辨法结合起来，就可直接同时求得任意感兴趣组分的纯光谱和纯色谱，从而可对该组分直接进行定性定量分析，而不必对所有组分进行全部分辨。这样的情况对那些含有大峰包小峰的分析体系特别有用，实因对那些含有大峰包小峰的分析体系目前还很难精确求解（参见下一节），致使不可能求得所有纯组分色谱。有关这样的具体例子参阅文献[179]。另外，由于该同时含于两个于矩阵的纯光谱 s 可以通过两种途径（$s=E\alpha$ 或 $s=D\beta$）而求得，这样我们可以对这两个所得结果进行比较，如这两个结果完全一样，说明所得结果可靠，反之，如所得结果相差大，则说明结果的可靠性不高，从而提供了一个检验分辨结果可靠性的工具。

(八) 二维色谱的一阶微分矩阵顺序秩分析方法

【基本思路】

在色谱数据中，经常出现小色谱峰完全被大色谱峰覆盖的情况，由于缺乏必要的信息，一般说来，是不能求得唯一解的，除非可以找到另外一些特殊条件。对于一般二维色谱数据分辨可求得唯一解的必要条件，Manne 给出了较详细的讨论，参阅文献[180]。本节所将要讨论的二维色谱的一阶微分矩阵顺序秩分析方法，关键就在于它利用了色谱峰局部对称的约束条件，引入微分色谱，才使得小色谱峰完全被大色谱峰覆盖的情况有可能得到唯一解，对于色谱峰局部不对称的体系，此法是有一定困难的。

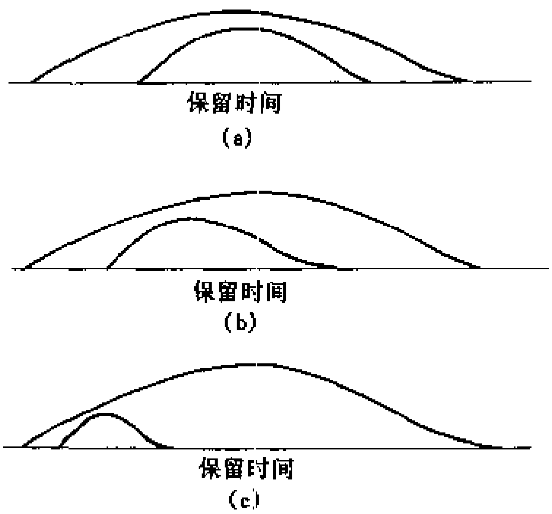


图 6-62　大色谱峰包小色谱峰的三种不同模式

对于小色谱峰完全被大色谱峰覆盖的二维色谱数据，其情况大致不外由图 6-62 所示的三种情况，在此三种情况中，本节所讨论的算法只适用第二种情况 (b)，即小峰的极大点与大峰的极大点不重合，而且在大峰的极大点处，小峰还未完全消失。在此三种情况中，第一种情况 (a) 因大小峰的极大点完全重合，故本法不适用；而对于第三种情况 (c)，则因在大峰的极大点处，小峰已完全消失，本法亦不适用。当然，当在大峰的极大点处小峰还未出现的情况与

第三种情况类似，亦不可采用此法。本法的基本思路就是要利用色谱峰有且只有一个极大点和在对极大点进行微分时其值为零的特点。

对于一大色谱峰包小色谱峰的二维色谱数据矩阵 $\boldsymbol{X}$，根据 Lambert-Beer 定律，可写成如下形式：

$$\boldsymbol{X}=\boldsymbol{c}_1\boldsymbol{s}_1^t+\boldsymbol{c}_2\boldsymbol{s}_2^t+\boldsymbol{E}$$

式中，$\{c_i, i=1,2\}$和$\{s_i^t, i=1,2\}$分别表示未知的两个组分的色谱和光谱，E 表示量测误差。这样的二维色谱分别是保留时间(t)和波长(w)的函数，所以，其中的每个元素 x_{ij}都是可写成如下形式

$$x_{ij}(t,w)=c_1(t)s_1(w)+c_2(t)s_2(w)+e_{ij}$$

即每个元素都是保留时间(t)和波长(w)的函数，对它们可以从两个方向来进行微分，如略去误差项，两个方向的微分式可分别表示为

$$\mathrm{d}x_{ij}(t,w)/\mathrm{d}t=s_1(w)\mathrm{d}c_1(t)/\mathrm{d}t+s_2(w)\mathrm{d}c_2(t)/\mathrm{d}t \tag{6-340a}$$

或

$$\mathrm{d}x_{ij}(t,w)/\mathrm{d}w=c_1(t)\mathrm{d}s_1(w)/\mathrm{d}w+c_2(t)\mathrm{d}s_2(w)/\mathrm{d}w \tag{6-340b}$$

因为二维色谱数据矩阵的每一行就是一个光谱，而每一列就是一个色谱，所以也可以将上述微分式(6-340)写成矢量形式，即

$$\mathrm{d}\boldsymbol{x}_i^t(t,w)/\mathrm{d}t=\boldsymbol{s}_1^t(w)\mathrm{d}c_1(t)/\mathrm{d}t+\boldsymbol{s}_2^t(w)\mathrm{d}c_2(t)/\mathrm{d}t \quad (i=1,\cdots,n) \tag{6-341a}$$

或

$$\mathrm{d}\boldsymbol{x}_j(t,w)/\mathrm{d}w=\boldsymbol{c}_1(t)\mathrm{d}s_1(w)/\mathrm{d}w+\boldsymbol{c}_2(t)\mathrm{d}s_2(w)/\mathrm{d}w \quad (j=1,\cdots,m) \tag{6-341b}$$

在此，$\{\mathrm{d}\boldsymbol{x}_i^t(t,w)/\mathrm{d}t,(i=1,\cdots,n)\}$表示二维色谱数据对时间方向的微分矩阵 $\mathrm{d}\boldsymbol{X}/\mathrm{d}t$ 的每一行，而$\{\mathrm{d}\boldsymbol{x}_j(t,w)/\mathrm{d}w,(j=1,\cdots,m)\}$则表示二维色谱数据对波长方向的微分矩阵 $\mathrm{d}\boldsymbol{X}/\mathrm{d}w$ 的每一列，此外，式(6-341)中隐含了一个十分重要的关系，即分别在两个纯组分色谱的极大点处，也就是说 $\mathrm{d}c_1(t)/\mathrm{d}t=0$ 或 $\mathrm{d}c_2(t)/\mathrm{d}t=0$ 的时候，可以很容易地分别得到两个物种的纯光谱，亦即

$$\mathrm{d}\boldsymbol{x}_i^t(t,w)/\mathrm{d}t|_{\mathrm{d}c_1(t)/\mathrm{d}t=0}=\boldsymbol{s}_2^t(w)\mathrm{d}c_2(t)/\mathrm{d}t \tag{6-342a}$$

和

$$\mathrm{d}\boldsymbol{x}_i^t(t,w)/\mathrm{d}t|_{\mathrm{d}c_2(t)/\mathrm{d}t=0}=\boldsymbol{s}_1^t(w)\mathrm{d}c_1(t)/\mathrm{d}t \tag{6-342b}$$

同理，在两个纯组分光谱的极大点处也就是说 $\mathrm{d}s_1(t)/\mathrm{d}t=0$ 或 $\mathrm{d}s_2(t)/\mathrm{d}t=0$ 的时候，可以很容易地分别得到两个物种的纯色谱，亦即

$$\mathrm{d}\boldsymbol{x}_j(t,w)/\mathrm{d}w|_{\mathrm{d}s_1(t)/\mathrm{d}w=0}=\boldsymbol{c}_2(t)\mathrm{d}s_2(w)/\mathrm{d}w \tag{6-343a}$$

和

$$\mathrm{d}\boldsymbol{x}_j(t,w)/\mathrm{d}w|_{\mathrm{d}s_2(t)/\mathrm{d}w=0}=\boldsymbol{c}_1(t)\mathrm{d}s_1^t(w)/\mathrm{d}w \tag{6-343b}$$

因为本法的基本思路就是要利用色谱峰有且只有一个极大点的特点，所以在以后的讨论中，我们将只讨论对时间方向的微分矩阵 $\mathrm{d}\boldsymbol{X}/\mathrm{d}t$ 的情况。而对时间方向的微分矩阵 $\mathrm{d}\boldsymbol{X}/\mathrm{d}t$ 的每一行仍是存在于体系中的两个光谱的线性组合[参见式(6-341)]，因此，经此微分以后，此矩阵的秩仍是等于 2，而且它的线性关系仍然是保持不变的，当然，式中的系数不再是非负的了。另外，根据式(6-342)，只要找到了两个纯组分的极大点，就可以得到两个纯物种的光谱。现在的问题是怎样才能找到这两点的位置，确定了这两点的位置，就等于找到了两个物种的纯光谱，这就是本法要解决的关键问题。

对于时间方向的微分矩阵 $\mathrm{d}\boldsymbol{X}/\mathrm{d}t$，在某一个纯物种的色谱极大点处，即 $t_{\max,i}(i=1,2)$，我们

有 $dc_i(t)/dt=0$。不失一般性，可以先讨论 $i=1$ 的情况，即第一个色谱极大点的情况，因为此时 $dc_1(t)/dt=0$，所以有

$$dx_i^t(t,w)/dt|_{t_{max,1}}=(dc_2(t)/dt)s_2^t(w)=\beta s_2^t(w) \tag{6-344}$$

即可得到，第二组分的纯光谱，如果在第一组分色谱极大点的附近色谱峰是近似对称的，则我们有

$$dc_1(t)/dt|_{t_{max-1,1}}=-dc_1(t)/dt|_{t_{max+1,1}} \tag{6-345}$$

利用式(6-345)所提供的条件，如果将微分矩阵 dX/dt 在第一组分色谱极大点的前一点和后一点的两个光谱相加，可以得到

$$\begin{aligned}
&dx_i^t(t,w)/dt|_{t_{max-1,1}}+dx_i^t(t,w)/dt|_{t_{max+1,1}}\\
&=\{s_1^t(w)dc_1(t)/dt+s_2^t(w)dc_2(t)/dt\}|_{t_{max-1,1}}+\{s_1^t(w)dc_1(t)/dt+s_2^t(w)dc_2(t)/dt\}|_{t_{max+1,1}}\\
&=[dc_1(t)/dt|_{t_{max-1,1}}+dc_1(t)/dt|_{t_{max+1,1}}]s_1^t+[dc_2(t)/dt|_{t_{max-1,1}}+dc_2(t)/dt|_{t_{max+1,1}}]s_2^t\\
&=[dc_2(t)/dt|_{t_{max-1,1}}+dc_2(t)/dt|_{t_{max+1,1}}]s_2^t\\
&=\mu s_2^t
\end{aligned} \tag{6-346}$$

在此利用了式(6-345)所提供的在第一组分色谱极大点的附近色谱峰是近似对称的条件。如果将由式（6-344）和式（6-346）所得的两个光谱组成一个子矩阵，亦即 $[\beta s_2^t, \mu s_2^t]$，则此子矩阵的秩为 1。

【基本算法】

根据以上讨论的这些结果，可以很容易地构造一个算法来确定二维微分色谱（对时间方向的）中两个纯组分的色谱极大点的位置了。该算法可由以下几个步骤来完成：

① 在对于时间方向的微分矩阵 dX/dt 中选择第一个行矢量，即 dx_i^t/dt；

② 第二个行矢量就将该矢量的前一个和后一个行矢量，即 dx_{i-1}^t/dt 和 dx_{i+1}^t/dt，相加而成；

③ 将这两个行矢量组成一个矩阵，并对它进行主成分分析以得到两个特征值；

④ 从微分矩阵 dX/dt 第二个行矢量开始，不断地重复上述①到③步，一直到该微分矩阵 dX/dt 的倒数第二个行矢量，这样，就可以得到一系列的特征值；

⑤ 将由此得到的一系列特征值对保留时间方向作图，在图上找到的第二个特征值曲线中出现的两个最小点的位置，也就是两个纯组分色谱的极大点的位置，这两点所代表的微分矩阵 dX/dt 的行矢量，也就是两个物种的纯光谱。

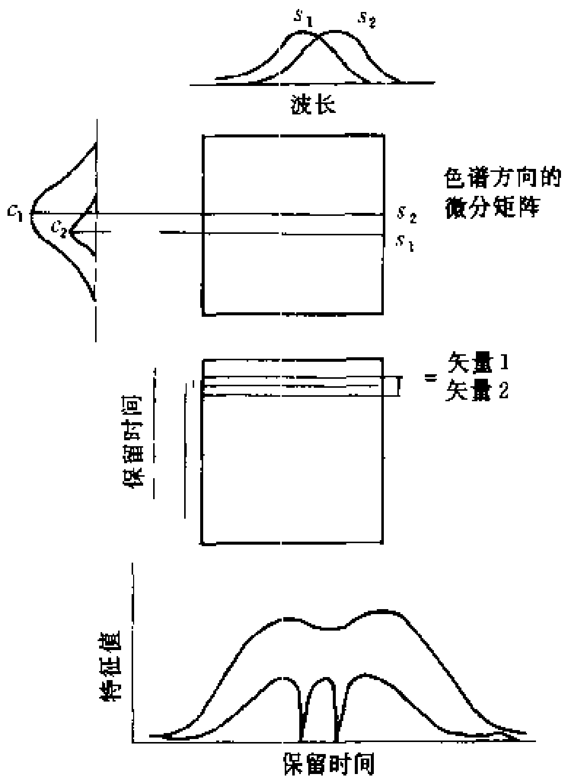


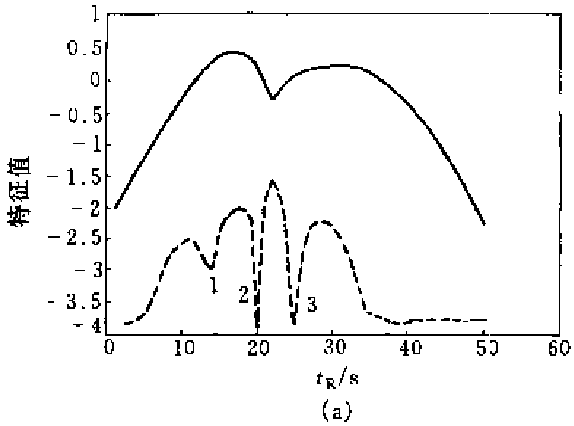
图 6-63 一阶微分矩阵顺序秩分析方法原理示意图

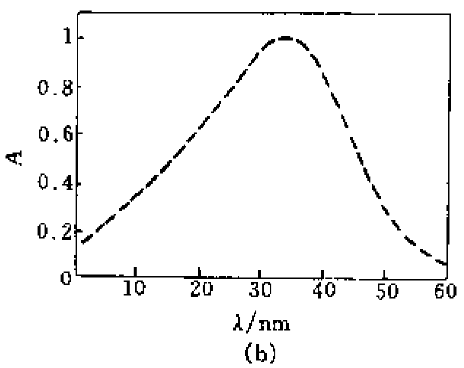
s_1, s_2 —两个纯组分的光谱；
c_1, c_2 —两个纯组分的色谱；
对原始二维色谱量测矩阵微分后，在两个色谱极大值处可得该两个组分的纯物种光谱（s_1 和 s_2）；继用顺序秩分析方法对该微分矩阵进行分析，可得图下部的秩分析结果，图中的两个极小点即为两个纯光谱所在点

【讨论】

此算法的构成基本思路和期望所得到的结果形象地示于图 6-63。如果在纯色谱峰的附近的

确是对称的，则此法可以得到很理想的结果如图 6-64 所示。图 6-64 示出的是一个用对称的高斯峰来模拟色谱峰的二组分体系的结果。另外，因此法只适用在图 6-62 所示三种情况的第二种，所以，在进行分辨之前，似有必要先确定我们碰到的情况是否的确属于图 6-62 所示的第二种情况。对于这三种情况的判别，可采用特征投影图，对于上述的三种不同情况，它们的特征投影图示于图 6-65。从图可以看出，对于这三种情况，它们的特征投影图都不同，而且各具特色，很容易分别。





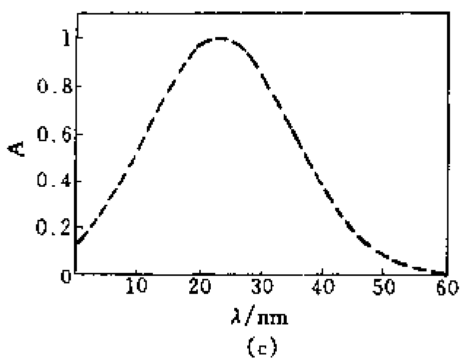


图 6-64　一个大峰包小峰模拟体系的顺序秩分析结果图

(a) 为秩分析图，图 (a) 上部三个极小点 (1，2，3)，其中第 2 和第 3 点处的秩为 1，应为相对应的两个纯物种光谱所在位置。根据此两点所得的纯光谱与真实纯光谱比较图示于图 (b) 和 (c)

----为估计光谱；——为实际纯组分光谱，从图中可以看出，估计值与实际值吻合甚好

然而，在很多情况下，在纯色谱峰附近的对称这一点前提很难满足，这就使该法的适用性受到很大限制。对此，该法是否有相应的解决办法呢？这是我们将要继续讨论的一点。

因为一般的色谱峰都不同程度的存在着拖尾情况，当拖尾情况存在时，纯色谱峰附近的对称这一点前提就很难保证，在此，用两个对称的高斯峰来组成一个色谱峰，使模拟体系的色谱具有拖尾现象。这样组成的两个大峰包小峰的二组分模拟体系示于表 6-17。对于由这样具有拖尾色谱体系用顺序秩分析方法所得结果示于图 6-66。从图 6-66 可以看出，得到的不是 2 个，而是 3 个 [图 6-66(a)] 或者 4 个 [图 6-66(b)] 极小点，而且这些极小点都还没有达到噪声水平，这时，我们应该用什么方法来确定两个纯色谱峰的极大点呢？

对于这样的情况，可以借助特征投影图来帮助决定两个纯色谱峰的极大点。从本章第一节的讨论可知，如果将一两组分的混合体系的光谱都标准化成具有总和皆等于某一常数，它们在主成分投影图的投影点就为一条直线，混合物光谱的点都应落在两个纯物种光谱的投影点之间，这样就可以利用这一特点来决定两个纯色谱峰的极大点，实因如果在顺序秩分析方

法所得的几个最小点为纯光谱，它们必须会将处于直线的两端，反之，它们就会落在混合物的光谱之中。基于这一推断，可以这样来构造一个新矩阵，首先将借助原始 X 标准化成具有总和为一常量的光谱，然后将由顺序秩分析方法所得的几个最小点也同样标准化组成一个新矩阵，继将它们进行主成分特征投影，如果，所得的几个最小点中有一个是处在投影图的端点，而其他几个在中间，则这一处于端点的就可能是最靠近纯物种的光谱点。对表 6-17 的两个模拟例子的计算结果示于图 6-67，它们与实际纯物种光谱的比较示于图 6-68。从图可知，这样所得的结果是很靠近实际结果的。所以，采取这样的方法有可能得到一个很靠近实际的近似解，不失为一个补救策略，对于该法的更详细讨论可参阅文献[181]。当然，继续研究如何得到在色谱峰附近不对称的分辨算法仍是具有重要意义的。

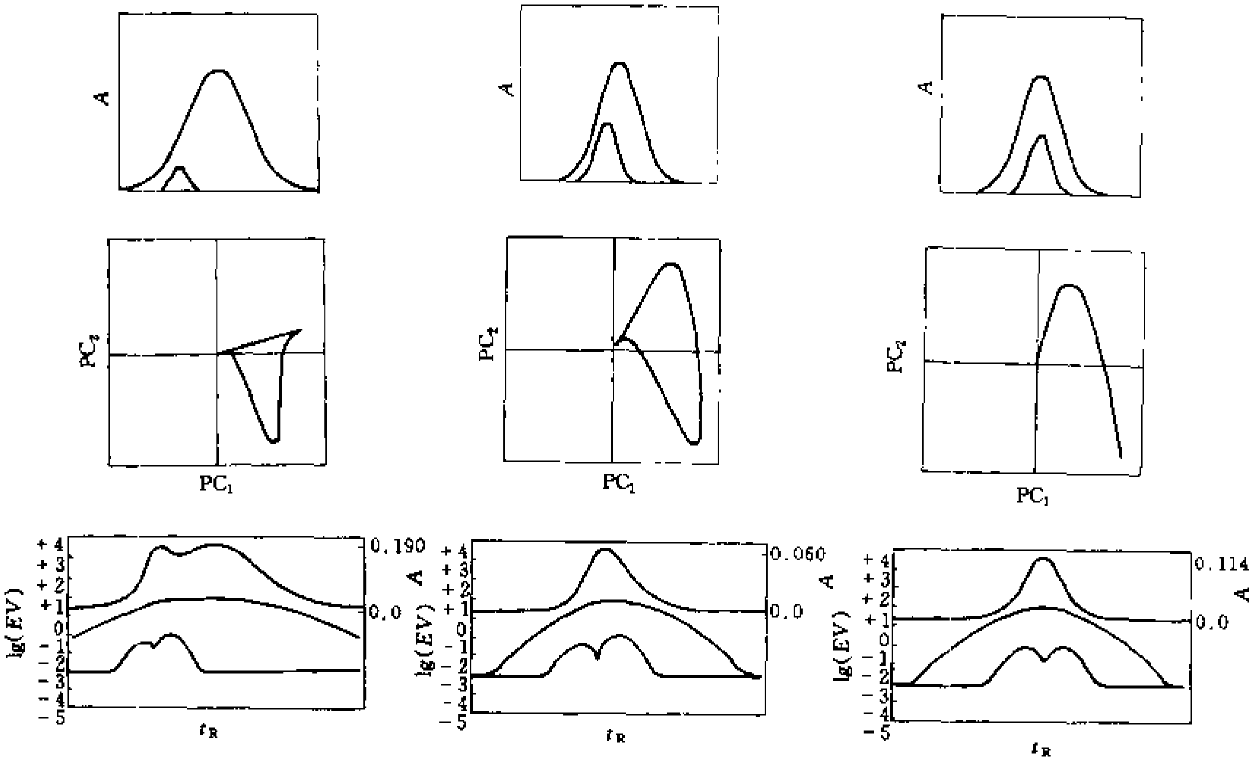


图 6-65　三种不同模式的大色谱峰包小色谱峰的特征投影图与特征结构跟踪图（ETA）结果

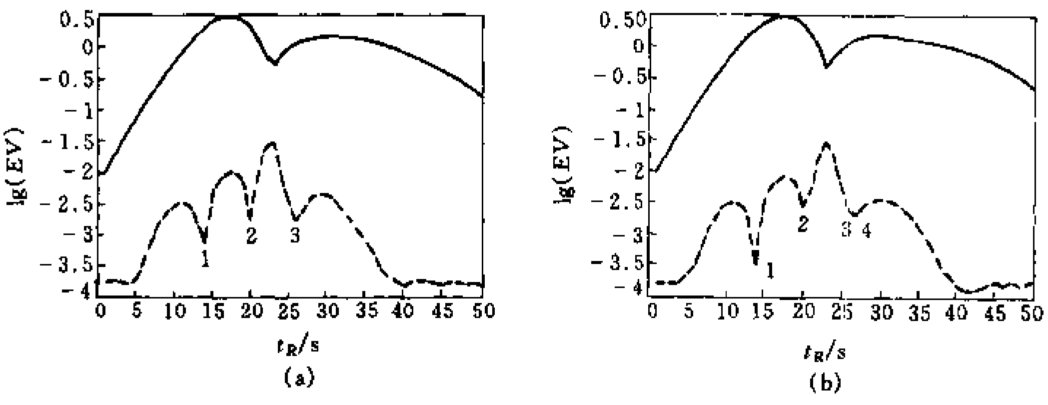


图 6-66　由表 6-17 所示的两个不对称大色谱峰包小色谱峰情况的一阶微分矩阵顺序秩分析结果图

1，2，3，4—极小值点

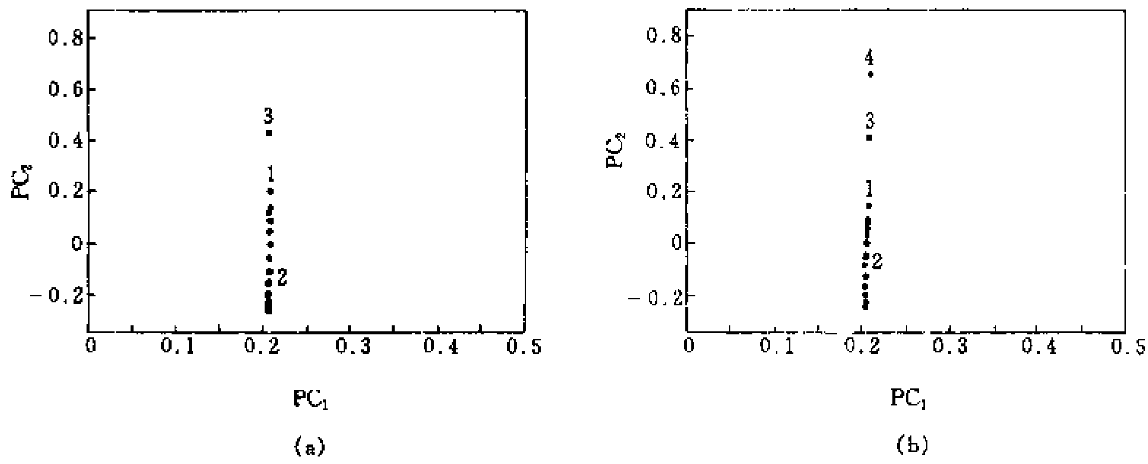


图 6-67 由图所对应之主成分投影图

(a) 图对应于图 6-66 中(a)图所示情况；(b) 图对应于图 6-66 中(b)图所示情况

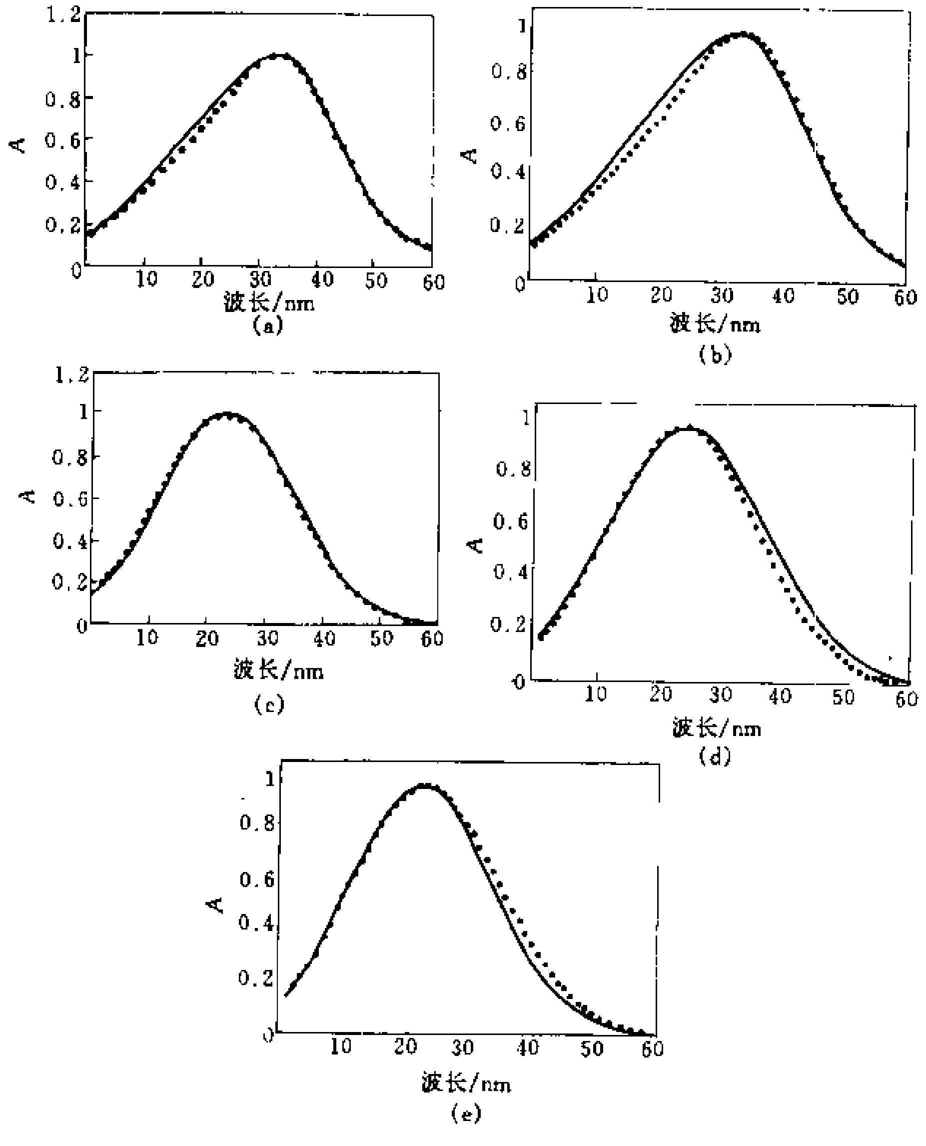


图 6-68 两个模拟拖尾体系的纯光谱估计比较图

……为估计值；　为实际值

(a)，(b)与图 6-66(a)图及图 6-67(a)图中的 2，3 点相对应的光谱；

(c)，(d)，(e)与图 6-66(a)图及图 6-67(b)图中的 2，3，4 点相对应的光谱

表 6-17 模拟拖尾色谱的参数表

色谱峰	大高斯峰			小高斯峰		
	峰位置	标准差	峰高	峰位置	标准差	峰高
大色谱峰	30	6.44	1.0	40	7.07	0.3
	30	6.44	1.0	40	7.07	0.1
小色谱峰	25	3.54	0.5	29	3.87	0.1
	25	3.54	0.5	29	3.87	0.2

三、张量分辨方法

正如前面所讨论的，对于黑色分析体系的解析，要获得唯一解，需要三维，即张量类型的数据。此节中将详细讨论这种张量分辨的两种主要方法，通过对这两种方法的讨论，将可以更清楚地看到,只有这种基于张量类型数据的分辨方法才能在任何没有附加的约束条件下，获得唯一解。

张量分辨方法在化学中还将是一类很有前途的方法,因在众多的分析化学所用的仪器中，联用仪器已成为一般分析化学实验室中的常用仪器，如高效液相色谱与二极管阵列检测器联用仪（HPLC-DAD），气相色谱与质谱联用仪（GC-MS），气相色谱与红外光谱联用仪（GC-IR），高效液相色谱与质谱联用仪（HPLC-MS），质谱与质谱联用仪（MS-MS），荧光发射光谱与激发光谱仪（EX-EM）等，而且，这种联用的趋势还在不断扩大，Hirschfeld 为此列出了一个可能被联用的分析仪器的一览表[182,183]，据 Hirschfeld 所列表来估计，在这些联用仪中，大约 60%可产生双线性数据，可以相信，这种联用趋势还将不断扩大。注意到将几个不同样本所测得的双线性数据合在一块就成了三线性的数据，所以，本章将要讨论的张量分辨方法将会成为很有用的分析方法。其实，现在就已经存在一些可产生三维数据的分析仪器，如三维核磁共振仪、相分辨荧光光谱仪（Phase-resolved fluorescene spectroscopy，PRFS），本身就可以产生三维数据，从这一角度考虑，本节所要介绍的方法可能很快将成为分析化学的常用方法。

在详细讨论张量分辨方法之前，先一般化地讨论一下张量分辨的数学模型。以下我们将采用黑花体来表示三维张量，如$\mathscr{A}$、$\mathscr{B}$、$\mathscr{C}$ 等，对于一个三线性的量测张量，一般可用式（6-347）表示。

$$\mathscr{A}=\boldsymbol{x}\otimes\boldsymbol{y}\otimes\boldsymbol{z}+\boldsymbol{\varepsilon} \tag{6-347}$$

式中，$\otimes$表示矢量间的外积，即 $\boldsymbol{x}\otimes\boldsymbol{y}=\boldsymbol{x}\boldsymbol{y}^t$；$\boldsymbol{\varepsilon}$ 表示与量测张量$\mathscr{A}$ 相对应的量测误差张量。

（一）投影旋转因子分析法

【基本思路】

投影旋转因子分析法（PRFA）是由 Burdick 等针对相分辨荧光光谱仪（PRFS）所产生的三维数据的解析而提出的[184]。其主要思路仍是基于因子分析法，不同的是它巧妙地利用了张量数据的性质来进行投影旋转，而获得有物理意义的唯一解。下面将对此法进行详细的介绍。

由相分辨荧光光谱仪（PRFS）所产生的数据可用下式表示。

$$\boldsymbol{M}_f=\sum \boldsymbol{x}_k\boldsymbol{y}_k^t d_{k,f}=\boldsymbol{X}\boldsymbol{D}_f\boldsymbol{Y}^t \tag{6-348}$$

式中，$\boldsymbol{M}_f$ 为在频率 f 下所产生的二维荧光光谱。其中 $\boldsymbol{x}_k$ 和 $\boldsymbol{y}_k$（$k=1,2,\cdots,n$）分别表示第 k 物种的荧光激发光谱和发射光谱矢量；$d_{k,f}=\alpha_k\cos\left[\tan^{-1}(\omega_k\tau_k)\right]\cos\left[\phi_D-\tan(\omega_k\tau_k)\right]$，在此 α_k 为一依赖于荧光物质浓度、荧光量子效率、分子摩尔吸光系数及仪器因素的常数；τ_k 是

第 k 物种的荧光寿命，ϕ_D 为检测器的相角。另外，在此 $X=[x_1, x_2, \cdots, x_n]$；$Y=[y_1, y_2, \cdots, y_n]$，$D_f$ 为一对角矩阵，它的第 k 个对角元素即为 $d_{k,f}$。相分辨荧光光谱仪就是通过调节不同的频率来得到具有不同频率的二维荧光光谱，以构成一个三维数据阵

$$g=\sum x_k \otimes y_k \otimes z_k \tag{6-349}$$

式中，$z_k=[d_{k,1}, d_{k,2}, \cdots, d_{k,F}]$，$F$ 为不同频率的数目。

式(6-349)这样的三维数据还可表示成

$$g_{ijf}=\sum (x_k)_i (y_k)_j d_{k,f} \tag{6-350}$$

式中，$(x_k)_i$ 表示荧光激发光谱 x_k 中的第 i 个元素；$(y_k)_j$ 表示荧光发射光谱 y_k 中的第 j 个元素。在此 $i=1, 2, \cdots, L$，即 L 个激发波长点；$j=1, 2, \cdots, M$；即 M 个发射波长点；$f=1, 2\cdots, F$，即 F 个不同调制频率，所以 $\boldsymbol{G}$ 是一个 $(L\times M\times F)$ 的三维数据阵。

黑色分析体系的分辨算法的目的就在于直接从量测三维数据 $\boldsymbol{G}$ 出发，分辨出纯组分的荧光激发光谱 $x_k(k=1, 2, \cdots, n)$，荧光激发光谱 $y_k(k=1, 2, \cdots, n)$ 以及与组分浓度有关的 $d_{k,f}(k=1, 2, \cdots, n)$ 来。投影旋转因子分析法（PRFA）的解析思路是首先通过多维尺度分析的 TUCKALS2 方法[185]求得不同 $\boldsymbol{M}_f$ 的共同列矢量空间 $\boldsymbol{U}$ 及其共同行矢量空间 $\boldsymbol{V}^t$，然后通过适当的投影和旋转技术，即可得到分辨的唯一解。以下将对其投影旋转过程给出详细介绍。

【基本算法】

投影旋转因子分析法解析的基本过程可由下述几步构成：

（1）投影运算过程　用共同行矢量空间 $\boldsymbol{U}$ 和共同列矢量空间 $\boldsymbol{V}$ 对不同 $\boldsymbol{M}_f$ 的进行投影，即

$$\begin{aligned}\boldsymbol{M}_f &= \boldsymbol{U}(\boldsymbol{U}^t\boldsymbol{U})^{-1}\boldsymbol{U}^t\boldsymbol{M}_f\boldsymbol{V}(\boldsymbol{V}^t\boldsymbol{V})^{-1}\boldsymbol{V}^t \\ &= \boldsymbol{U}\boldsymbol{C}_f\boldsymbol{V}^t\end{aligned} \tag{6-351}$$

从式（6-351）可以看出 $\boldsymbol{C}_f$ 是一个 $(n\times n)$ 阶的方阵。

（2）旋转分辨过程　因为 $\boldsymbol{U}$ 矩阵的列向量与 $\boldsymbol{X}$ 矩阵的列向量张成同一线性空间，且 $\boldsymbol{V}^t$ 矩阵的行向量与 $\boldsymbol{Y}^t$ 矩阵的行向量张成同一线性空间，所以，总可以找到两个 $(n\times n)$ 阶的满秩方阵 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$，使得

$$\boldsymbol{X}=\boldsymbol{U}\boldsymbol{P} \text{ 和 } \boldsymbol{Y}=\boldsymbol{V}\boldsymbol{Q} \tag{6-352}$$

将式（6-352）代入（6-348）式

$$\boldsymbol{M}_f=\boldsymbol{U}\boldsymbol{P}\boldsymbol{D}_f\boldsymbol{Q}^t\boldsymbol{V}^t \tag{6-353}$$

继与式（6-351）比较，即 $\boldsymbol{M}_f=\boldsymbol{U}\boldsymbol{C}_f\boldsymbol{V}^t=\boldsymbol{U}\boldsymbol{P}\boldsymbol{D}_f\boldsymbol{Q}^t\boldsymbol{V}^t$，所以有

$$\boldsymbol{C}_f=\boldsymbol{P}\boldsymbol{D}_f\boldsymbol{Q}^t \tag{6-354}$$

对式（6-354）的两边分别左乘 $\boldsymbol{P}^{-1}$ 及右乘 $(\boldsymbol{Q}^t)^{-1}$，立即可得

$$\boldsymbol{D}_f=\boldsymbol{P}^{-1}\boldsymbol{C}_f(\boldsymbol{Q}^t)^{-1}$$

一般说来，$\boldsymbol{P}$ 与 $\boldsymbol{Q}$ 可通过对 $\boldsymbol{C}_i\boldsymbol{C}_j^{-1}$ 或通过对 $\boldsymbol{C}_i^{-1}\boldsymbol{C}_j$ 的正交分解来得到，即

$$\boldsymbol{P}^{-1}\boldsymbol{C}_i\boldsymbol{C}_j^{-1}\boldsymbol{P}=\boldsymbol{D}_i\boldsymbol{D}_j^{-1} \tag{6-355a}$$

或

$$\boldsymbol{Q}^t\boldsymbol{C}_i^{-1}\boldsymbol{C}_j(\boldsymbol{Q}^t)^{-1}=\boldsymbol{D}_i^{-1}\boldsymbol{D}_j \tag{6-355b}$$

这是因为

$$\boldsymbol{D}_i\boldsymbol{D}_j^{-1}=\boldsymbol{P}^{-1}\boldsymbol{C}_i(\boldsymbol{Q}^t)^{-1}[\boldsymbol{P}^{-1}\boldsymbol{C}_j(\boldsymbol{Q}^t)^{-1}]^{-1}$$

$$=P^{-1}C_i(Q^t)^{-1}Q^tC_j^{-1}P=P^{-1}C_iC_j^{-1}P \qquad (6\text{-}356)$$

这就是（6-355a）式，同理，（6-355b）式亦可依此推出。得到 $D_i^{-1}D_j$ 或 $D_iD_j^{-1}$ 后，再利用进行正交分解时得到的 P 与 Q 根据（6-352）式即可求出荧光激发光谱 X 或荧光发射光谱 Y 来。

（3）求取共同列矢量空间 U 及共同行矢量空间 V^t 的多维尺度分析过程　在投影旋转因子分析法中，求得不同 M_f 的共同列矢量空间 U 及其共同行矢量空间 V^t 的多维尺度分析的 TUCKALS2 方法实质上是一种交替式最小二乘法，此法主要由以下步骤所组成：

① 先求出共同列矢量空间 U 或共同行矢量空间 V^t 的初始解，在此，不失一般性，我们先求出共同列矢量空间 U 的初始解，即对 $\sum M_fM_f^t$ 进行主成分分解，以求得其 n 个最大特征值所对应的 n 个特征向量来组成初始 $U^{(0)}$。

② 以下为一交替迭代过程，即利用所得的 U 来求 V，然后，再利用所得的 V 来求新的 U，如此循环，直至收敛。

即对于 $s=1，2，\cdots$

a）令 $P_U^{(s)}=U^{(0)}(U^{(0)t}U^{(0)})^{-1}U^{(0)t}$，对 $\sum M_f^tP_U^{(s)}M_f$ 进行主成分分解，以求得其 n 个最大特征值所对应的 n 个特征向量来组成 $V^{(s)}$；

b）令 $P_V^{(s)}=V^{(s)}(V^{(s)t}V^{(s)})^{-1}V^{(s)t}$，对 $\sum M_fP_V^{(s)}M_f^t$ 进行主成分分解，以求得其 n 个最大特征值所对应的 n 个特征向量来组成 $U^{(s)}$；

c）如果所求得的 $\|U^{(s-1)}-U^{(s)}\|$ 小于某一给定值则停止迭代，否则转向 a 步。

在上述交替式迭代过程中，收敛是很快的，实因在每步的求解特征值和特征向量的过程中，就相当于是对 $\mathrm{trace}[P_V^{(s)}(\sum M_f^tP_U^{(s)}M_f)P_V^{(s)}]$ 或者 $\mathrm{trace}[P_U^{(s)}(\sum M_fP_V^{(s)}M_f^t)P_U^{(s)}]$ 的不断极大化过程，而 $\mathrm{trace}(\sum M_f^tM_f)$ 或 $\mathrm{trace}(\sum M_fM_f^t)$ 就分别是这一极大化过程的收敛点。

由上述迭代过程求出的共同列矢量空间 U 或共同行矢量空间 V^t 后，即可利用它们来进行分辨了。这样的一个过程，也是一个不断摈除噪声的过程，这一点在本章开头讨论主成分分析误差理论时就进行过详细的介绍。

（二）广义秩消失因子分析法

关于三线性分解的广义秩消失因子分析法是由 Sanchez 和 Kowalski 提出[186]。其基本思路仍与广义秩消失因子分析法大致相同，即利用广义特征值和特征向量的方法来求解三维分辨问题。在此，它将不局限于只像前一章所讨论的广义秩消失因子分析法仅用两个矩阵来进行求解，而将被拓广至多个矩阵进行直接分解。实际上，早在 20 世纪 70 年代，心理计量学的研究者就指出，对于三维数据，在无任何附加条件下因子分析法就可能求得唯一解[187,188]，得到唯一分解的必要条件是此数据应为一个三线性数据，即

$$g=\sum x_k\otimes y_k\otimes z_k$$

在三维数据的研究领域中，特别是在心理计量学的研究领域中，一般都把上式称为 PARAFAC-CANDECOMP 模型。PARAFAC-CANDECOMP 模型对上式求解是通过一个交替最小二乘过程来完成的，即 ALS 先在设已得到 $\{x_k\}$ 和 $\{y_k\}$ 的基础上去求 $\{z_k\}$，然后在设已得到 $\{y_k\}$ 和 $\{z_k\}$ 的基础上去求 $\{x_k\}$，再在设已得到 $\{z_k\}$ 和 $\{x_k\}$ 的基础上去求 $\{y_k\}$，如此重复，直至收敛。注意到在此 $\{x_k\}$、$\{y_k\}$ 和 $\{z_k\}$ 都是表示一组向量，即 $\{x_k\}$ 表示 $\{x_1,x_2,\cdots,x_n\}$，其中 n 为体系的组分数。Deligny 等首先将上述迭代 ALS 三线性分解法用于估计色谱柱参数与色谱保留时间的关系[189]；Appellof 和 Davidson 将另一三线性分解法，或所谓的 Tucker-1 法用于分析化学的色谱/激发/发射之三维数据[190]，Wold 等也将 Tucker-1 法用于其他的化学数据，如化学模式识别等[191]，关于这些三维数据的处理方法在文献[186]中也有简要介绍，下面主要介绍

Sanchez 和 Kowalski 提出的三线性分解的广义秩消失因子分析法。

1. 三维张量是由两个同样大小的矩阵 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 所组成时的三线性分解的广义秩消失因子分析法

【基本思路】

最简单的三维张量是由两个同样大小的矩阵 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 所组成，不失一般性，设式 $\boldsymbol{g}=\sum \boldsymbol{x}_k \otimes \boldsymbol{y}_k \otimes \boldsymbol{z}_k$ 中的 $\{z_k\}$ 张成张量的第三维，对于这最简单的情况，此三线性模型可表为

$$\boldsymbol{M}_1=\sum_{k=1}^{n} z_{k1}\boldsymbol{x}_k \otimes \boldsymbol{y}_k \tag{6-357a}$$

$$\boldsymbol{M}_2=\sum_{k=1}^{n} z_{k2}\boldsymbol{x}_k \otimes \boldsymbol{y}_k \tag{6-357b}$$

式中，n 为体系的组分数。可以看出，$\boldsymbol{z}_k=(z_{k1},z_{k2})$。对于黑色分析体系，我们的目的就在于求得 $\{\boldsymbol{x}_k\}$ 和 $\{\boldsymbol{y}_k\}$，求得 $\{\boldsymbol{x}_k\}$ 和 $\{\boldsymbol{y}_k\}$ 后，就相当于求得了各纯物种的标准激发与发射荧光光谱或是纯物种的标准色谱和光谱，它们的相对浓度即可 $\{\boldsymbol{z}_k\}$ 从中得到。下面将详细讨论广义秩消失因子分析法是怎样来求得 $\{\boldsymbol{x}_k\}$ 和 $\{\boldsymbol{y}_k\}$ 的。

设 $\{\boldsymbol{y}_k\}$ 是一组线性无关的矢量，则可以找到一个这样的矢量 $\boldsymbol{y}^r$，使得

$$\boldsymbol{y}_k^t\boldsymbol{y}^r=\delta_{kr} \tag{6-358}$$

式中，δ_{kr} 为 Kronecker 符号，即当 $k=r$ 时，$\delta_{kr}=1$，而当 $k\neq r$ 时，$\delta_{kr}=0$。这样的 $\boldsymbol{y}^r$ 在张量代数中被称为 $\{\boldsymbol{y}_k\}$ 这组基的第 r 个协变矢量。如果 $Y=[y_1,y_2,\cdots,y_n]$ 这样一个矩阵，那么，$\boldsymbol{Y}^r=[\boldsymbol{y}^1,\boldsymbol{y}^2,\cdots,\boldsymbol{y}^n]^t$，即是一个由 n 个协变矢量 $\{\boldsymbol{y}^r\}$ 为其行所构成的矩阵，则此时 $\boldsymbol{Y}^r$ 可看成是 $\boldsymbol{Y}$ 矩阵的一个广义逆，因，$\boldsymbol{Y}^r\boldsymbol{Y}=\boldsymbol{I}$，$\boldsymbol{I}$ 为一单位矩阵。如果将协变矢量 $\boldsymbol{y}^r$ 右乘式(6-69)的两边，立即可得

$$\boldsymbol{M}_1\boldsymbol{y}^r=z_{r1}\boldsymbol{x}_r \tag{6-359a}$$

$$\boldsymbol{M}_2\boldsymbol{y}^r=z_{r2}\boldsymbol{x}_r \tag{6-359b}$$

将式(6-359)两式合并，有

$$\boldsymbol{M}_1\boldsymbol{y}^r z_{r2}=\boldsymbol{M}_2\boldsymbol{y}^r z_{r1} \tag{6-360}$$

式 (6-360) 实际上是一个广义特征值-特征向量问题，对于这样的问题，因涉及同时将对角化问题，目前有效的算法是 QZ 算法，然而，从现有的计算数学发展的情况看，只有当 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 都是方阵时 QZ 算法方能求解，所以，广义秩消失因子分析法的重要步骤首先就是将 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 都预先转换成为方阵。

将 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 都转换成为方阵面又不丢失信息的方法一般可以采用同时主成分分解来实现。实际上，使用本节（一）讨论的多维尺度分析的 TUCKALS2 方法也是可以实现将 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 都转换成为方阵面又不丢失信息，但为保持介绍广义秩消失因于分析法的原始性，在这里将 Sanchez 和 Kowalski 提出的算法给出详细介绍。

【基本算法】

（1）降维过程　此降维过程是通过求得该两矩阵的共同列向量或行向量空间的基向量继用投影而完成的。下面将分别给予简要介绍。

① 求取共同列向量和行向量空间的基向量过程　因为 $\boldsymbol{M}_i=\sum z_{ki}\boldsymbol{x}_k \otimes \boldsymbol{y}_k=\sum z_{ki}\boldsymbol{x}_k\boldsymbol{y}_k^t$，在此 $i=1$，2，即只有两个同样大小的矩阵，所以，如果想要求得与 $\{\boldsymbol{x}_k\}$ 张成同一空间的一组正交基 $\{\boldsymbol{u}_k\}$，可将 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 按列进行扩展，即令

$$\boldsymbol{M}=[\boldsymbol{M}_1|\boldsymbol{M}_2]$$

也就是说，如果 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 是分别都是一个$(L\times \boldsymbol{M})$的阶的矩阵的话，那么，$\boldsymbol{M}$ 矩阵就是一个

$(L\times 2M)$阶的矩阵，这种方法在线性代数中称为展开方法。此时如果对 $\boldsymbol{M}$ 进行奇异值分解的话，就可以取其得分矢量以得到与$\{\boldsymbol{x}_k\}$张成同一空间的一组正交基$\{\boldsymbol{u}_k\}$，即

$$\boldsymbol{M}=\boldsymbol{USV}^t$$

$\{\boldsymbol{u}_k\}$将与$\{\boldsymbol{x}_k\}$张成同一空间。同理也可将 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 从另一个方向进行展开，即令

$$\boldsymbol{M}=\begin{bmatrix}\boldsymbol{M}_1\\ \cdots\\ \boldsymbol{M}_2\end{bmatrix}$$

也就是说，如果 $\boldsymbol{M}_1$ 和 M_2 是分别都是一个$(L\times M)$的阶的矩阵的话．那么，M 矩阵就是一个$(2L\times \boldsymbol{M})$阶的矩阵。这样，就可对 $\boldsymbol{M}$ 进行奇异值分解后取其载荷矢量以得到与$\{\boldsymbol{y}_k\}$张成同一空间的一组正交基$\{\boldsymbol{v}_k\}$。

② 投影降维过程　采用如此得到的正交基$\{\boldsymbol{u}_k\}$和$\{\boldsymbol{v}_k\}$所构成的矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 来对 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 进行投影运算，即可得到相应的不丢失任何信息的方阵，具体可依式(6-361)进行

$$\boldsymbol{M}_{1p}=\boldsymbol{U}^t\boldsymbol{M}_1\boldsymbol{V} \tag{6-361a}$$

及

$$\boldsymbol{M}_{2p}=\boldsymbol{U}^t\boldsymbol{M}_2\boldsymbol{V} \tag{6-361b}$$

在此-$\boldsymbol{M}_{1p}$和 $\boldsymbol{M}_{2p}$分别为 $\boldsymbol{M}_1$ 和 M_2 在正交基$\{\boldsymbol{u}_k\}$和$\{\boldsymbol{v}_k\}$所构成的矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 的投影之下所得到的两个方阵，这是因为 $\boldsymbol{U}$ 矩阵是一个$(L\times n_a)$阶矩阵，而 $\boldsymbol{V}$ 矩阵则是一个$(M\times n_a)$阶矩阵，n_a 在此表示我们在用正交基$\{\boldsymbol{u}_k\}$和$\{\boldsymbol{v}_k\}$构成矩阵 U 和 V 时所取的主成分的个数．所以，$\boldsymbol{M}_{1p}$和 M_{2p}分别为两个$(n_a\times n_a)$阶方阵。为不丢失原始矩阵 $\boldsymbol{M}_1$ 和 M_2 的信息，一般可将 n_a 取大点，即大于分析体系的实际组分数，以保证在以后的广义特征值-特征向量问题求解的 QZ 算法可顺利进行。另外，细心的读者可能会提出，既然上述所得的正交基$\{\boldsymbol{u}_k\}$和$\{\boldsymbol{v}_k\}$分别与$\{\boldsymbol{x}_k\}$和$\{\boldsymbol{y}_k\}$构成同一线性子空间，那么，M_{1p}和 $\boldsymbol{M}_{2p}$会不会就成为对角矩阵?由对 M_1 和 $\boldsymbol{M}_2$ 分别直接进行奇异值分解方法所得的正交基也是与$\{\boldsymbol{x}_k\}$和$\{\boldsymbol{y}_k\}$构成同一线性子空间，而那样所得的正交基是可以分别将 $\boldsymbol{M}_1$ 和 M_2 对角化的，不同在什么地方，留给读者自己去仔细思考。

(2) QZ 分辨过程　将 M_{1p}和 $\boldsymbol{M}_{2p}$分别代替式(6-360)中的 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$，然后进行 QZ 算法计算，即可得 $\boldsymbol{y}^r$ 和 z_{r1}/z_{r2}，为求得纯物种的标准化的激发光谱$\{\boldsymbol{x}_k\}$，由式(6-359a)及(6-359b)，可得

$$\boldsymbol{x}_r=\boldsymbol{M}_1\boldsymbol{y}^r/\parallel \boldsymbol{M}_1\boldsymbol{y}^r\parallel$$

同理，如果对 M_1^t 和 $\boldsymbol{M}_2^t$ 进行上述过程，即可求出纯物种的标准化的激发光谱$\{\boldsymbol{y}_k\}$来，这是因为像式(6-360)一样，可得式(6-362)

$$\boldsymbol{M}_1^t\boldsymbol{x}^r z_{r2}=\boldsymbol{M}_2^t\boldsymbol{x}^r z_{r1} \tag{6-362}$$

【讨论】

以上已将三线性分解的广义秩消失因子分析法对最简单的张量分辨的情况作出了介绍。它虽与在灰色分析体系中讨论的广义秩消失因子分析法有相似的地方，但已知条件和求解的目标是不同的。对于灰色分析体系中讨论的广义秩消失因子分析法，它的重点在于定量，即在于求 $z_k,(k=1,2,\cdots,n)$，然而，在本节所讨论三线性分解的广义秩消失因子分析法，因对$\{\boldsymbol{x}_k\}$和$\{\boldsymbol{y}_k\}$一无所知，其重点首先就在于定性，即在于求纯物种的标准激发光谱$\{\boldsymbol{x}_k\}$和标准发射光谱$\{\boldsymbol{y}_k\}$，从这个意义说来，它们又是不同的，当然，解题的思路也不同了。

另外，在此还值得强调指出的是，虽然三线性分解的广义秩消失因子分析法并没有什么特

殊假设，但是，该法仍存在一些约束条件，这对于正确使用该法将十分重要，所以在这里有必要给出介绍。

① 对于由式(6-360)和式(6-362)所给出的无约束广义特征值-特征向量的问题，有可能出现成对的复数特征值，而这在本文所论及的校正和分辨的领域中是没有意义的，需要作出相应变换，转成实数解。Li 和 Gemperline 对此进行了详细研究，读者可参阅文献[73]。

② 从式(6-360)和式(6-362)所给出的广义特征值-特征向量的问题可以看出，凡属于 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 的零空间的任何向量 $\boldsymbol{e}$ 都能以任意特征值满足上述两式，实因此时有 $\boldsymbol{M}_2\boldsymbol{e}=\boldsymbol{0}$。如果得到了这样的解，必须除去才行。我们可以采用式(6-363)来检验所得的解是否属于 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 的零空间矢量，如果

$$\boldsymbol{M}_2\boldsymbol{y}^r\approx\boldsymbol{0} \tag{6-363}$$

成立，则说明所得的解为 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 的零空间的向量，必须除去。同时，也可以采用前述的主成分投影方法来避免所得的解落入 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 的零空间和降低噪声影响。

③ 三线性分解的广义秩消失因子分析法虽对于数据无特殊要求，但为求得有真实物理意义的解，它仍要求 $\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 都是双线性矩阵以及矢量集 $\{\boldsymbol{x}_k\}$ 和 $\{\boldsymbol{y}_k\}$ 为线性无关矢量。

④ 对于三线性数据的浓度矢量 $z_k(k=1,2,\cdots,n)$，广义秩消失因子分析法要求它们不能对应成比例，即它们不能是指向同一方向的矢量，亦即必须有 $z_k^t z_s\leqslant|z_k\|z_s|$。如果这一条件不能满足，它将导致所得结果不可靠。

2. 三维张量是由 D 个同样大小的矩阵 $\boldsymbol{M}_k$ 所组成时的三线性分解的广义秩消失因子分析法

【基本思路】

将有 D 个双线性矩阵，即

$$\boldsymbol{M}_1=\sum_{k=1}^{n}z_{k1}\boldsymbol{x}_k\otimes\boldsymbol{y}_k \tag{6-364a}$$

$$\boldsymbol{M}_2=\sum_{k=1}^{n}z_{k2}\boldsymbol{x}_k\otimes\boldsymbol{y}_k \tag{6-364b}$$

$$\vdots\qquad\qquad\vdots$$

$$\boldsymbol{M}_D=\sum_{k=1}^{n}z_{kD}\boldsymbol{x}_k\otimes\boldsymbol{y}_k \tag{6-364c}$$

$z_k=(z_{k1},z_{k2},\cdots,z_{kD}),(k=1,2,\cdots,n)$。

这样，可以得到 $D(D-1)/2$ 个广义特征值-特征向量的求解问题，即

$$\boldsymbol{M}_1\boldsymbol{y}^r z_{r2}=\boldsymbol{M}_2\boldsymbol{y}^r z_{r1} \tag{6-365a}$$

$$\boldsymbol{M}_1\boldsymbol{y}^r z_{r3}=\boldsymbol{M}_3\boldsymbol{y}^r z_{r1},\boldsymbol{M}_2\boldsymbol{y}^r z_{r3}=\boldsymbol{M}_3\boldsymbol{y}^r z_{r2} \tag{6-365b}$$

$$\vdots\qquad\vdots\qquad\vdots\qquad\vdots\qquad\vdots$$

$$\boldsymbol{M}_1\boldsymbol{y}^r z_{r2}=\boldsymbol{M}_D\boldsymbol{y}^r z_{r1},\boldsymbol{M}_2\boldsymbol{y}^r z_{rD}=\boldsymbol{M}_D\boldsymbol{y}^r z_{r2},\cdots,\boldsymbol{M}_{D-1}\boldsymbol{y}^r z_{r2}=\boldsymbol{M}_D\boldsymbol{y}^r z_{rD-1} \tag{6-365d}$$

对于由式(6-365)所表出的广义特征值-特征向量的求解问题，由于量测噪声的存在，是很难找到一个像前述只含两个矩阵的那种简单情况那样的单个 $\boldsymbol{y}^r$ 来直接求解 $D(D-1)/2$ 个广义特征值-特征向量问题的，而且，对于这样多个广义特征值-特征向量的求解问题，还不能保证其中任意两个矩阵都存在共同的矢量集 $\{\boldsymbol{x}_k\}$ 和 $\{\boldsymbol{y}_k\}$。因此，对于由上述式子表出的广义特征值-特征向量的求解问题需另辟新径。

与前述只单纯从两个矩阵来考虑广义特征值-特征向量问题求解的思路相反，我们可以找

两个可代表所有 D 个矩阵的新矩阵，即它们的线性组合来解决这一问题。令

$$\boldsymbol{S}_p = \sum_{k=1}^{D} w_{pk}\boldsymbol{M}_k \tag{6-366}$$

式中，$p=1,2$，所以，$\boldsymbol{S}_p$ 为两个与原来 D 个 $\boldsymbol{M}_k(k=1,2,\cdots,D)$ 矩阵相同大小，但是它们所有矩阵的线性组合而成的矩阵，$w_{pk}(p=1,2;k=1,2,\cdots,D)$ 为其线性组合系数，它们的正确选定将成为多个广义特征值-特征向量求解问题的广义秩消失因子分析法的关键，因将式(6-366)展开可得

$$\begin{aligned}\boldsymbol{S}_p &= \sum_{k=1}^{D} w_{pk}\boldsymbol{M}_k = \sum_{k=1}^{D} w_{pk}\sum_{r=1}^{n} z_{rk}\boldsymbol{x}_r \otimes \boldsymbol{y}_r \\ &= \sum_{r=1}^{n}\boldsymbol{x}_r \otimes \boldsymbol{y}_r \sum_{k=1}^{D} w_{pk}z_{rk}\end{aligned}$$

令 $a_{pr}=\sum w_{pk}z_{rk}=\boldsymbol{w}_p^{t}\boldsymbol{z}_r$，则有

$$\boldsymbol{S}_p = \sum_{r=1}^{n}\boldsymbol{x}_r \otimes \boldsymbol{y}_r \sum_{k=1}^{D} w_{pk}z_{rk} = a_{pr}\left(\sum_{r=1}^{n}\boldsymbol{x}_r \otimes \boldsymbol{y}_r\right) \tag{6-367}$$

从式(6-367)可以看出，$a_{pr}(p=1,2;r=1,2,\cdots,n)$ 最好都不为零方可使 $\boldsymbol{S}_p$ 能成为所有 $\sum \boldsymbol{x}_r\otimes\boldsymbol{y}_r$ 的线性组合。Sanchez 和 Kowalski 提出的三线性分解的广义秩消失因子分析法是通过下述方法来求得 $w_{pk}(p=1,2;k=1,2,\cdots,D)$ 的。

从图 6-69 可以看出，任意一个三线性张量可以从三个方向进行展开，下面将用展开的方法来求 $w_{pk}(p=1,2;k=1,2,\cdots,D)$。先将由 D 个相同大小矩阵 $\boldsymbol{M}_k(k=1,2,\cdots,D)$ 按 y 方向进行展开，这样可得一个$(D\times ML)$阶的矩阵，将其记为 M_D，对其进行主成分分析后取其两个对应于最大特征值的得分矢量来作为 $\boldsymbol{w}_p(p=1,2)$，这样取得的 $\boldsymbol{w}_p(p=1,2)$可满足 $\boldsymbol{w}_p^t\boldsymbol{z}_k\neq 0$ 的条件，以保证可将由 n 个相同大小矩阵 $\boldsymbol{x}_r\otimes\boldsymbol{y}_r(r=1,2,\cdots,n)$的信息都包括在 $\boldsymbol{S}_p$ 中。当然，求 $\boldsymbol{w}_p$ $(p=1,2)$不局限于这一种方法，别的方法也是可行的。只要能将 n 个相同大小矩阵 $\boldsymbol{x}_r\otimes\boldsymbol{y}_r(r=1,2,\cdots,n)$的信息都包括在 $\boldsymbol{S}_p$ 中即可。

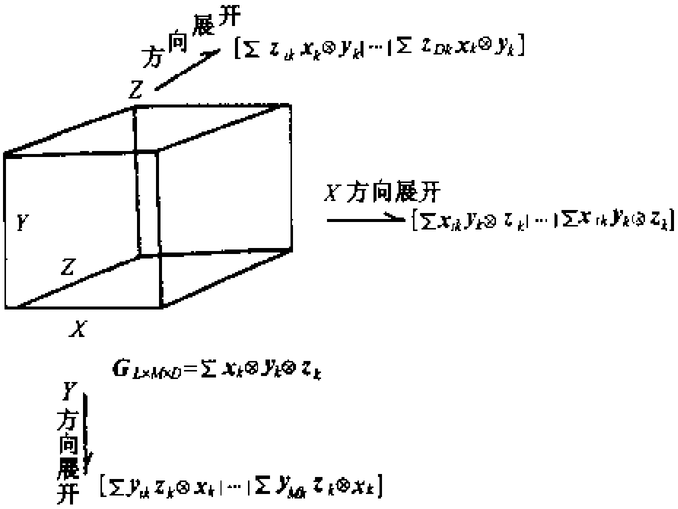


图 6-69　三线性张量的三个方向展开示意图

【基本算法】

由 D 个同样大小的矩阵 $\boldsymbol{M}_k$ 所组成张量的三线性分解广义秩消失因子分析法的具体算法由如下步骤构成：

(1) 用Tucker-3 模型将一个阶数为$(L\times M\times D)$三维张量$\mathscr{A}=\sum\boldsymbol{x}_k\otimes\boldsymbol{y}_k\otimes\boldsymbol{z}_k$ 转化成为两

个$(n\times n)$阶的矩阵M_1和M_2。在此，L为激发光谱的波长数；M为发射光谱的波长数；D为不同样本的量测所得的二维光谱矩阵的个数；n为体系化学组分数(实际上它是未知的)。

① 将三维张量$\mathscr{M}$按z方向展开(参见图6-69)，得到一个$(L\times MD)$阶矩阵后，对其进行主成分分解，继取其中n个得分矢量构成一个与矢量集$\{x_k\}$张成同一线性子空间的矩阵U：

$$[M_1|\cdots|M_D]=U_1S_1V_1^t \qquad 并令 \quad U=U_1(L\times n)$$

式中，U为一$(L\times n)$阶矩阵，它收集了前n个得分矢量。

② 将三维张量$\mathscr{M}$按x方向展开(参见图6-69)，得到一个$(M\times LD)$阶矩阵后，对其进行主成分分解，继取其中n个得分矢量构成一个与矢量集$\{y_k\}$张成同一线性子空间的矩阵V：

$$[M_1^t|\cdots|M_D^t]=U_2S_2V_2^t \quad 并令 \quad V=U_2(L\times n)$$

式中，V为一$(M\times n)$阶矩阵，它收集了前n个得分矢量。

③ 将三维张量$\mathscr{M}$按y方向展开(参见图6-69)，得到一个$(D\times ML)$阶矩阵后，对其进行主成分分解，继取其中两个得分矢量构成一个与矢量集$\{z_k\}$张成同一线性子空间的矩阵W：

$$[\sum y_{1k}z_k\otimes x_k|\cdots|\sum y_{1k}z_k\otimes x_k]=U_3S_3V_3^t \quad 并令 \quad W=U_3(D\times 2)$$

在此，W为一$(D\times 2)$阶矩阵，它收集了前两个得分矢量。

④ 将三维张量$\mathscr{M}$投影于$\{\mathscr{U},\mathscr{V},\mathscr{W}\}$为基的特征空间之上，以得到两个$(n\times n)$阶的矩阵$M_1$和$M_2$，

$$M_1=\sum_{k=1}^{D}w_{pk}(U^tM_kV),M_2=\sum_{k=1}^{D}w_{pk}(U^tM_kV)$$

(2) 对两个$(n\times n)$阶的矩阵M_1和M_2来进行广义特征值-特征向量问题求解

① 使用QZ算法以得到满足下式的特征向量x^r，

$$M_1x^r\lambda_{2r}=M_2x^r\lambda_{1r}$$

② 将所得的n个特征向量$x^r(r=1,2,\cdots,n)$收集以构成一个矩阵，继求得它的逆，记为X

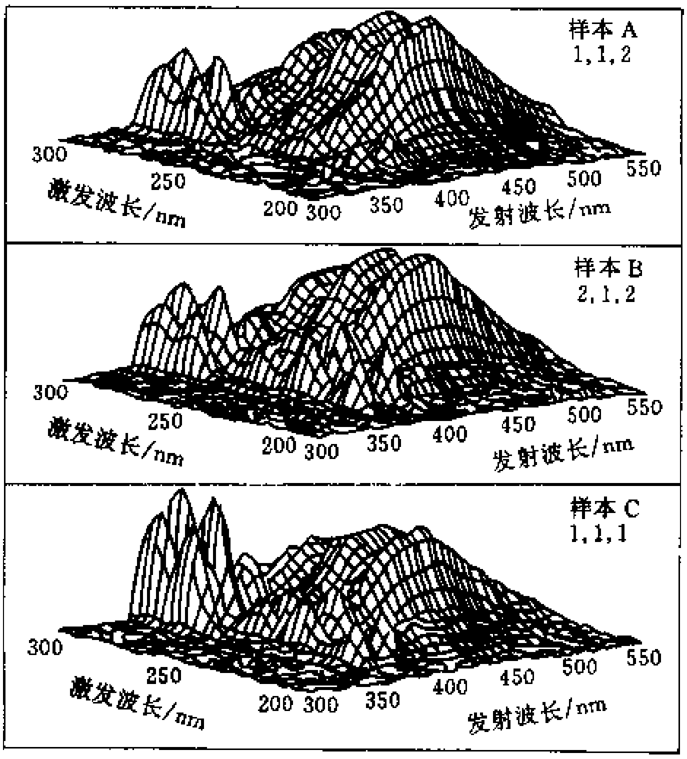


图6-70 三个模拟混合体系的二维光谱图

③ 使用 QZ 算法以得到满足下式的特征向量 y^r，

$$M_1^t y^r \lambda_{2r} = M_2^t y^r \lambda_{1r}$$

④ 将所得的 n 个特征向量 $y^r(r=1,2,\cdots,n)$收集以构成一个矩阵，继求得它的逆，记为 $\boldsymbol{Y}$

(3) 求纯物种的激发光谱$\{x_r\}$和发射光谱$\{y_r\}$及其相对浓度$\{z_r\}$

① $\boldsymbol{X}=\boldsymbol{UX}$ 在此 $\boldsymbol{X}$ 矩阵收集了 n 个矢量$\{x_r\}$；

② $\boldsymbol{Y}=\boldsymbol{VY}$ 在此 $\boldsymbol{Y}$ 矩阵收集了 n 个矢量$\{y_r\}$；

③ 用求得的激发光谱$\{x_r\}$和发射光谱$\{y_r\}$来求相对浓度$\{z_r\}$，即对 D 个二维量测谱 $\boldsymbol{M}_k(k=1,2,\cdots,D)$进行回归运算：

$$\boldsymbol{M}_k = z_{k1}\boldsymbol{x}_1 \otimes \boldsymbol{y}_1 + \cdots + z_{kn}\boldsymbol{x}_n \otimes \boldsymbol{y}_n \qquad (k=1,2,\cdots,D)$$

以求得$\{z_r\}$。

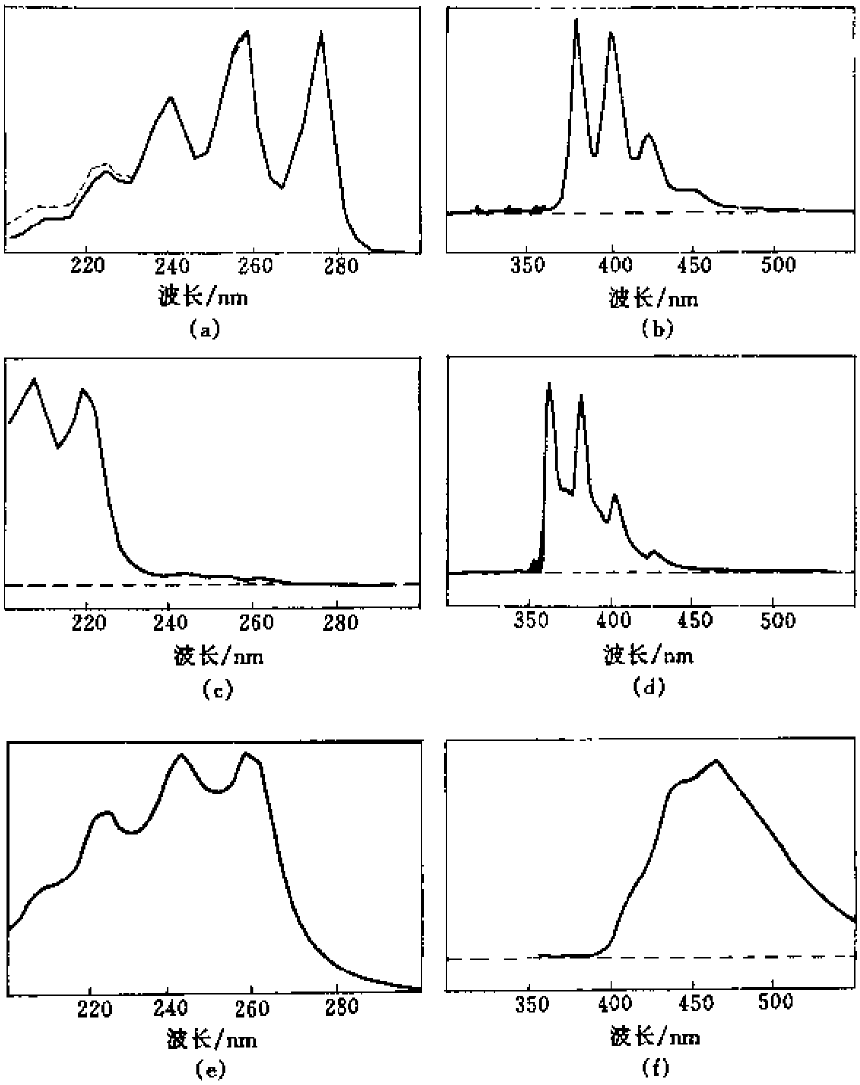


图 6-71 采用全部样本的二维荧光光谱之线性组合的广义秩消失因子分析法的计算所得结果

(a) 蒽激发光谱；(b) 蒽发射光谱；(c) 萘激发射光谱；(d) 萘发射光谱；(e) 荧蒽激发光谱；(f) 荧蒽发射光谱

【讨论】

Sanchez 和 Kowalski 用广义秩消失因子分析法对一个模拟的三组分荧光二维光谱体系进

行了计算，他们模拟的三个样本分别含不同浓度的蒽、䓛和荧蒽，它们的相对含量分别为：样本A为(1,1,2)；样本B为(2,1,2)；样本C为(1,1,1)。三个样本的二维荧光光谱图分别示于图6-70。在每个二维荧光光谱中均加入了1%的正态分布的量测误差。

图6-71示出了从三线性直接分辨的广义秩消失因子分析法，即采用全部样本的二维荧光光谱之线性组合的计算方法的所得结果，从图6-71可以看出，所得结果是相当令人满意的。图6-72示出了只用样本A和B来计算的广义秩消失因子分析法的计算结果，因为两个样本的相对浓度比A/B为(0.5,1,1)，即和荧蒽的相对浓度比例是相同的，所以，此时的广义秩消失因子分析法的计算结果对于这两种物种是不令人满意的。

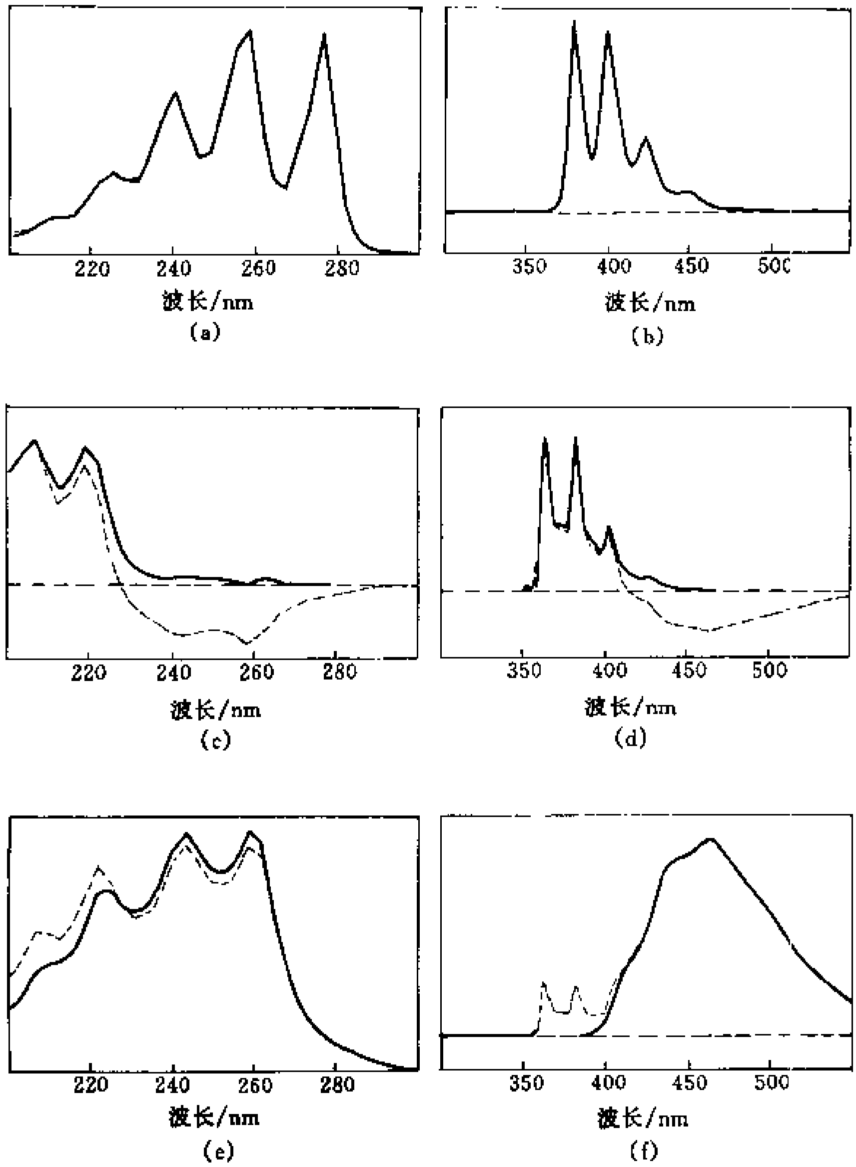


图6-72 只采用样本A和B的二维荧光光谱来计算的广义秩消失因子分析法的结果

(a) 蒽激发光谱；(b) 蒽发射光谱；(c) 䓛激发光谱；

(d) 䓛发射光谱；(e) 荧蒽激发光谱；(f) 荧蒽发射光谱

此外，还值得提出的是，在广义秩消失因子分析法的计算中，对于组分数的估计也是一个十分重要的步骤，从Sanchez和Kowalski的经验，对组分数的估计过多对结果影响不大，多估计的组分所对应的是噪声谱，很容易甄别，但是，如果对组分数估计不够，即小于体系实际组分数时，对结果影响很大，而致使所得结果不可靠。

参考文献

1 国家自然科学基金委员会．自然科学学科发展战略调研报告·分析化学．北京：科学出版社，1993
2 Lorber A. Anal Chem，1986；58：1167
3 Booksh，K S，Kowalski B R. Anal Chem，1994；66：762A
4 俞汝勤．"化学计量学与现代分析化学"，见张焘主编"科学前沿与未来"，第二集．北京：科学出版社，1996；198～208
5 Sachez E，Kowalski B R. J Chemometrics，1990；2：247
6 Sachez E，Kowalski B R. J Chemometrics，1990；2：265
7 Faber K，Lorber A and Kowalski B R. J Chemoemtrics，1997；11：419
8 Lorber A，Faber K and Kowalski B R. Anal Chem，1997；69：1620
9 Faber K and Kowalski B R. J Chemoemtrics，1997；11：181
10 Junker A and Bergmann G. Fresennius'Z. Anal Chem，1976；278：191
11 Thijssen P C，Kateman G and Smit H C. Anal Chim Acta，1984；157：99
12 Thijssen P C，Vogels L J P，Smit H C et al. Fresennius'Z. Anal Chem，1985：320：531
13 Sasaki K，S. Kawata and S. Minami，Applied Spectroscopy，1986，40：185
14 Liang Y Z，Xie YL and Yu R Q，Anal Chim Acta，1989；222：347～357
15 Junker A，Bergmann G and Fresennius'Z. Anal Chem，1974；272：267
16 Kaiser H，Fresennius'Z. Anal Chem.，1972；260：252
17 Otto M and Wegscheider W. Anal Chim Acta，1986；180：445
18 Rutan S，Chemom and Intell Lab Sys，1989；6：191
19 Brown S D. Anal Chim Acta，1986；181：1
20 Mok K W and Chau F T. Trends in Analytical Chemistry，1996；15：170～174
21 王柯敏，李洁，俞汝勤．高等学校化学学报，1989；10：356
22 Liu Y M，Yu R Q. Tananta，1988；35：707
23 Brown C W，Lynch P F，Obermski R J，et al. Anal Chem，1982：54：1472～1479
24 Kisner H J，Brown C W，and Kavernos G J. Anal Chem，1982；54：1479～1485
25 Brown C W，Lynch P F，Obermski R J，et al. Anal Chem，1983；55：1703～1707
26 Dief A S. Advanced Matrix Theory for Scientists and Engineers，Abacus Press，Tunbridge Wells & London，1982：93
27 Stewart G W，Introduction to Matrix Computations. New York：Academic Press，1973：340
28 俞汝勤．化学计量学导论，长沙：湖南教育出版社，1991：94
29 Hoskuldsson A. J Chemometrics，1988；2：211～228
30 Manne R. Chemom and Intell Lab Sys，1987；2：257～266；Martin H，Neas T. Multivariate Calibration，Wiley，Chichester，1989
31 Saxbery B E H，Kowalski B R. Anal Chem，1979；51；1038
32 Sharaf M A，Illman D L，Kowalski B R. Chemometrics. New York：John Willey & Sons，1986：139
33 Clancey V J. Nature 1947；159：339～340
34 Harris E K，DeMets D L. Clin Chem，1972；18；605
35 Barnett V，Lewis T. Outlier in Statistical data，3rd edn.，Wiley，Chichester，1993
36 Edgeworth F Y. On observations relating to several quantities，Hermathena 1887；6：279～285
37 Rousseeuw P J. Tutorial to robust statistics，Journal of Chemometrics，1991：5：1～20
38 王松桂．近代回归分析．上海：上海科技出版社，1988
39 Hodges J L. Proc. Fifth Berkeley Symp Math Stat Probab. 1967；1：163～168
40 Hampel H R. Ann. Math. Stat，1971：42：1887～1896
41 Rousseeuw P J，Leroy A M. Robust regression and outlier detection. John Wiey & Sons，1987
42 Liang Y Z，Fang K T. The Analyst，1996；121：1025～1029
43 Wagner H M. J Amer. Statist. Assoc. 1959；54：206
44 Barrodle I，Roberts F D K. SIAM J Num Anal，1973；10：839

45 王继红,湖南大学硕士研究生学位论文,1994
46 Huber P J. Robust Statistics, New York:John Wiley & Sons,1981
47 Andrews, Technometric,1974;16:523
48 Holland P W, Welsch R E. Commun Stat (Theory and Methods),1977;6:813~828
49 Philips G R, Eyring E R. Anal Chem,1983;55:1134~1138
50 Wolters R. and Kateman G. J Chemometrics,1989;3:329~342
51 Wei W Z,Zhu W H,Yao S Z. Chemom and Intell Lab Sys,1993;18:17~26
52 Xie Y L, Liang Y Z, Jiang J H et al. Anal Chim Acta,1995;311:185
53 Rousseeuw P J. J Am Stat Assoc. 1984;79:871~880
54 Steele J M,Steiger W L. Discrete applied Mathematics,1986;14:93~100
55 Massart D L. Kaufman L,Rousseeuw P J,et al. Anal Chim Acta,1985;187:171
56 Rutan et al. Anal Chim Acta, 1988;215:131
57 Ukkelberg A, Borgen O S. Anal Chim Acta,1993;277:489~494
58 Walczak B,Massart D L. Chemom and Intell Lab Sys,1994;277:489~494
59 Devlin J S,Gnanadesikan R,Kettenring J R,J Am Statistical Association, 1981;76:354~362
60 Wakeling I N, Macfie H J H. J Chemometrics, 1992;6:189~198
61 Beaton A E, Tukey J W. Technometrics, 1974;16:147
62 Stevens J P. Psychol Bull,1984;95:334~344
63 Cook R D,Weisberg S. Residuals and influence in regression, Chapman & Hall London, 1982
64 Hocking R R. Technometrics,1983;25:219~249
65 Cook R D. Technometrics, 1977;19:15~18
66 Rousseeuw P J,van Zomeren B C. J Ame Stat Association, 1990;85:633~639
67 Rousseeuw P J. "Multivariate estimation with high breakdown point," in Mathematical statistics and applications, (Vol. B, eds Grossmann, W, Pflug G,Vincze I Dordrecht:Reidel Publishing,1985:283~297
68 Atkinson A C Mulira H M. Statistics and Computing, 1993;3:27~35
69 Naes T Chemom and Intell Lab Sys, 1989;5:155~168
70 Hu Y Z,Smeyyers-verberk J,Massart D L. Chemom and Intell Lab Sys, 1990;9:31
71 陈文灿,崔卉,梁逸曾. 高等学校化学学报,1995;16:1349
72 Huber P J,The Annals of statistics, 1985;13: 435~475
73 Li S,Gemperline P J. Anal Chem,1992;64:599
74 Friedman J H,Stuetzle W. J Ame Stat Association, 1981;76:817~823
75 Friedman J H,Stuetzle W. J Ame Stat Association, 1984;79:599~608
76 Li G Y,Chen Z L. J Ame Stat Association, 1985;80:759~766
77 Xie Y L,Wang J H,Liang Y Z et al. J Chemometrics, 1993; 7: 527~541
78 Ammann L P,J Ame Stat Association, 1994;88:505~514
79 Frank I E. Chemom and Intell Lab Sys, 1995;27:1
80 Widrow B. Hoff M E. WESCON convention record. Part 4 (Aug. 1960), p99
81 Hopfield J J. Proc Natl Acad Sci U S A, 1982;79:2554;1984;81:3088
82 Hopfield J J, Tank D W. Biol Cybern, 1985;52:141
83 Rumelhart D E, McClelland J L,et al. Parallel distributed processing: explorations in the microstructure of cognition. Cambridge M A: MIT press, 1986
84 Lippmann R P. Proc 1987 IEEE International Conference on Neural Networks. San Diago,1989;5:417~426
85 Lippmann R P. Neural Communication, 1989;1:1~33
86 Wieland A Leighton R. IEEE First International Conference on Neural Networks, 1987;3:385~392
87 刘平,梁逸曾,王素国等. 化学学报,1997;54;386
88 宋新华,陈茁,俞汝勤. 中国科学(B辑),1993;23:245
89 Simits J R M, Melssen W J,Buydens L M C et al. Chemom and Intell Lab Sys, 1994;23:267

90 史忠福,神经计算. 北京:电子工业出版社,1993

91 Simits J R M, Melssen W J,Buydens L M C et al. Chemom and Intell Lab Sys, 1994:22:165

92 张承福. 力学进展,1994;24:186

93 焦李成. 神经网络计算. 西安:西安电子科技大学出版社,1993

94 梁逸曾,卢志强,俞汝勤. 高等学校化学学报,1989;10:704

95 Kateman G,Pijpers F W. Quality control in analytical chemistry , Wiley 1981;79

96 Correy W P, Kowalski R B. Anal Chem, 1986;58:3077

97 Currie L. A. Anal. Chem. ,1968;40:586

98 Zscheble F P,Murray H C,Baker G A et al. Anal. Chem, 1962;34:1776

99 Lorbor A. Anal Chem, 1985;57:2397~2399

100 梁逸曾,谢玉珑,俞汝勤,高等学校化学学报,1990;16:34~45

101 Malinowaski E R, Howery D G. Factor analysis in chemistry, New York:Wiley,1980

102 梁逸曾,谢玉珑,俞汝勤,科学通报,1989;34:1533~1538

103 Rutan S C,Brown S D. Anal Chim Acta,1984;160:99

104 Rutan S C,Brown,S D. Anal Chim Acta,1985;167:23

105 Karstang T,Kvalheim O M. Anal Chem, 1991;63:767~772

106 Ho C N,Christian G D,Davidson E R. Anal Chem,1978;50:1108~1113

107 Ho C N,Christian G D,Davidson E R. Anal Chem,1980;52:1071~1079

108 Ho C N,Christian G D,Davidson E R. Anal Chem,1981;53:92~98

109 McCue M, Malinowski E D. Journal of Chrmotographic Science,1983; 21:229~234

110 Burns D H,Callis J B,Christian C D. Anal Chem,1986,58:2805~2811

111 Lorber A. Anal. Chim. Acta,1984;164:164

112 Sanchez E,Kowalski B R. Anal Chem, 1986;58:499~501

113 梁逸曾,谢玉珑,俞汝勤. 化学学报,1990;48:472~477

114 Xie Y L,Liang Y Z,Yu R Q. Anal Chim. Acta,1993;272:61~72

115 Xie Y L,Liang Y Z,Yu R Q. Anal Chim. Acta,1993;276:233~244

116 格罗布 G H,万罗安 C F. 矩阵计算. 大连:大连理工大学出版社, 1988

117 Lorbor A. Anal Chem 1984;56:1004~1010

118 Malinowski E R, Howery D G. Factor analysis in Chemistry. New York: Wiley, 1980

119 Sanchez E ,Ramos L S,Kowalski B R. Journal of Chromatography, 1987;385:151~164

120 Ramos L S, Sanchez E,Kowalski B R. Journal of Chromatography,1987;385:165~180

121 Wilson B E,Sanchez E,Kowalski B R. J Chemometrics, 1989;3:493

122 Ohman J,Geladi P, Wold S. J Chemometrics, 1990;4:79~90

123 Ohman J,Geladi P, Wold S. J Chemometrics, 1990;4:135~144

124 Liang Y Z, Manne R,Kvalheim O M. Chemom and Intell Lab Sys,1992;14:175~184

125 Burton K W C, Nickness O. Chemom and Intell Lab Sys, 1987;1:135~149

126 Himmelblau,D M. Applied Nonlinear Optimization. New York:McGraw-Hill, 1972

127 Avriel M. Nonlinear programming:Analysis and Method. Englewood Cliffs, NJ:Prentice Hall,1976

128 Kirkpatrick S,Gelatt C D,Vecchi M P. Science, 1983;220;671

129 Bohachevsky I O,Johnson M E,Sein M L. Techmometrics, 1986;28:209

130 Holland J H. Adaptation in Natural and Artificial Systems. Ann Arbor, MI:University of Michigan Press,1975

131 David L. Handbook of genetic Algorithm. New York:Van Nostrand Reinhold, 1991

132 Xie Y L,Liang Y Z,Yu R Q. J Chemometrics,1993;7:369~379

133 陈文灿,陈增萍,许静等. 化学学报,1996;55:693~702

134 Bulmer T Shurvell H F. J Phys Chem, 1973;77:256

135 Battlett M S. Br. J Psychol Stat Seet,1950;3:77

136 Hugus Z Z,El-Awady Jr A A. J Phys Chem, 1971;75:2954

137 Kankare J J. Anal Chem, 1970;42:1322

138 Cattell R B. Educ Psych Meas,1958;18:791

139 Cochran R N, Horne F H. Anal Chem, 1977;49:846

140 Kowalski B R. Chemometrics: Theory and Application. ACS Symp. Ser. 52. Washington D. C: American Chemical society, 1977

141 Malinowski E R,Howery D G. Factor Analysis in Chemistry. New York: Wiley-Interscience,1980

142 Malinowski E R. J Chemometrics,1988;3:49

143 Lawton W E,Sylvester E A. Technometrics,1971;13:617～633

144 Osten D W,Kowalski B R. Anal Chem,1984;56:991～995

145 Sharaf M A,Kowalski B R. Anal Chem,1982;54:1291～1296

146 Kawata S,Komeda H,Sasaki K et al. Applied Spectrosc, 1985;39:610～614

147 Gillette P C,Lando J B, Koenig J L. Anal Chem, 1983;55:630～633

148 Ohta M. Anal Chem, 1973;45:553～556

149 Sharaf M A. Anal Chem,1986;58:3084

150 Borgen O S,Kowalski B R Anal Chim Acta,1985;174:1～26

151 Borgen O S,Davidson N, Mingyang Z,et al. Mikrochimica Acta, 1986;2:63～73

152 Gemperline P J. J Chem inf and Comput Sci 1984;24:206～212

153 Vanteginste B G M,Esser T, Bosman T,et al. Anal Chem,1985;57:971～985

154 Xu J, Liang Y Z,Yu R Q,J Chemometrics, 1995;10:63～76

155 梁逸曾,谢玉珑,俞汝勤,化学学报,1991;49:394～399

156 Zubbuhler A D,Maeder,M. Talanta,1986;181:287

157 Maeder M,A. D. Zubbuhler,Anal Chim Acta,1986;181:287

158 Maeder M. Anal Chem,1987;59:527

159 Maeder M,Zilian A. Chemom and Intell Lab Sys 1988;3:205

160 Keller H R,Massart D L. Anal Chim Acta,1991;246:279

161 Keller H R,Massart D L. Liang Y Z et al. Anal Chim Acta,1992;263:125

162 Malinowski E R. J Chemometrics,1992;6:29～40

163 Gemperline P J,Hamilton J C. J Chemometrics,1989;3:455

164 Kvalheim O M,Liang Y Z. Anal Chem, 1992;64:936～945

165 Liang Y Z,Kvalheim O M,Keller H R,et al. Anal Chem,1992;64:946～953

166 Geladi P, Wold S. Chemom and Intell Lab Sys, 1987;2:273～281

167 Liang Y Z,Kvalheim O M,Rahmani A,et al. Journal of Chemometrics,1993;7:15～43

168 Liang Y Z,Kvalheim O M,Rahmani A,et al. Chemom and Intell Lab Sys,1993;18:265～279

169 Liang Y Z,Kvalheim O M,Hoskulsson A. Journal of Chemometrics,1993;7:277～290

170 Liang Y Z,Kvalheim O M,Chemom and Intell Lab Sys,1993;20:115～125

171 Brereton R G,Rahmani A,Liang Y Z et al. Photochem, Boichem,1994;59:99～110

172 Libnau F O,Toft J, Christy A A et al. J Ame Chem Soc, 1994;116:8311～8316

173 沈海林,梁逸曾,俞汝勤等. 中国科学(B辑),1997;27:556～561

174 Lorber A. Anal Chem, 1986;58:1167

175 Liang Y Z,Kvalheim O M. Anal Chim Acta,1994;276:425～440

176 Liang Y Z,Kvalheim O M. Anal Chim Acta,1994;292:5～15

177 Xu Q S and Liang Y Z. Chemome and Intell Lab Sys 1999;45:335

178 Manne R,Shen H L,Liang Y Z. Chemom and Intell Lab Sys,1999;45:171

179 Shen H L, Liang Y Z,Manne R. Chemom and Intell Lab Sys 1999;45:323

180 R. Manne. Chemometrics and Intelligent Laboratory Systems,1995;27:89～94

181 Liang Y Z,Manne R,Kvalheim O M,Chemom and Intell Lab Sys,1994;22:229～240

182 Hirschfeld T. Anal Chem, 1980;52:297A

183 Hirschfeld T. Science, 1985; 230
184 Burdick D S, Tu X M, McGown LB, et al. J Chemometrics, 1990; 4: 15～28
185 Kroonenberg P M, De Leeuw J. Psychometrika, 1980; 45: 69
186 Sanchez E, Kowalski B R. J Chemometrics, 1990; 4: 29～45
187 Harshman R A. UCLA Working papers in phonetics, 1970; 16: 1
188 Carroll J D, Chang J J. Psychometrika, 1970; 35: 283
189 De Ligny C L, Spanjer M C, Van Houwelingen J C, et al. J Chromatogr. 1984; 301: 311
190 Appellof C J, Davidson E R. Anal Chem, 1981; 53: 2053～2056
191 Wold S, Geladi P, Esbensen K, et al. J Chemometrics, 1987; 1: 41

第七章　化学模式识别

化学模式识别（Chemical Pattern Recognition）是化学计量学研究中的一个十分重要的内容，它从化学量测数据（包括由多元校正和多元分辨取得的定性定量成分数据）出发，进一步揭示物质的隐含性质，为分析化学家提供了十分有用的决策性信息。模式识别的计算过程可有如下步骤：根据实验得来的一批训练点，参照化学（或物理）模型或经验规律提出一批特征量，然后进行进一步特征抽取，以求得合适的特征量，张成模式空间或特征空间（feature space）。必要时，对数据进行预处理，预处理后，即可通过模式识别算法进行训练和分类，然后根据训练（或称学习）分类所得的判据对未知样本进行判别（或称计算机预报）。这一过程可用图 7-1 框图表示。

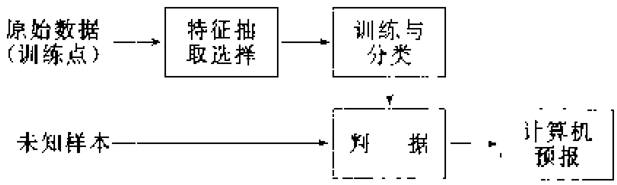


图 7-1　模式识别计算过程示意图

本章将对这几个步骤分别进行介绍。在介绍了模式空间的几种距离与相似性度量、特征抽取方法及模式识别的数据预处理方法的基础上，将对目前在化学计量学中用得最多的三大类化学模式识别方法，即所谓的有监督的模式识别方法、无监督的模式识别方法和特征投影的模式识别方法都给出详细的介绍，而且，各类方法的介绍将以其基本思路和基本算法为主线，并尽量给出相应的计算机程序。在近年来的化学计量学的研究中，人工神经网络和一些基于全局最优算法的分类方法受到了广大化学计量学家的密切关注，对于这些化学计量学中的模式识别新方法，本章亦将作出介绍。

第一节　模式空间的几种距离与相似性度量

1. 化学模式空间

任何一个化学样本或化工过程可用一组参量 $\boldsymbol{x}_i$（矢量）来表征，即 $\boldsymbol{x}_i=(x_{i1}, x_{i2}, \cdots, x_{in})$，在化学模式识别的文献中，这种参量值又称特征。在许多情况下，这些参量实际上就是原始化学量测数据。这样的一组 n 个参量就构成了化学模式识别空间的一个点，即一个 n 维的化学模式。由这样的化学模式所构成的 n 维空间，称化学模式空间，简称模式空间。一般说来，因高维模式空间提供了更多的信息，故有可能解决一些低维空间中难于解决的问题。这种情况可容易通过图 7-2 说明。从图 7-2 中可以看出，在一维空间、即沿 x_1 轴或 x_2 轴，单用 x_1 或 x_2 均无法区分 A 或 B，但在二维模式空间，可以由一条直线对 A 和 B 加以区分。一般而言，仅增加一个参量，难有如此显著的改善，但由一维空间（化学家最熟悉

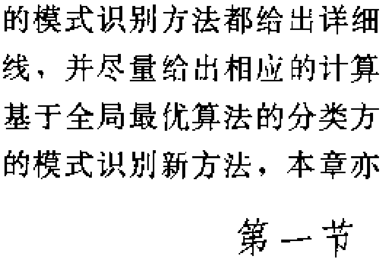


图 7-2　在二维空间可能区分一维空间不可分的二类样本

的）过渡到 n 维空间，n 越大，改善程度愈大，则是普遍的规律。这也是化学模式识别近年来受到化学家们重视的根本原因。

2. 模式空间的距离与相似性度量

化学模式识别的基本假设是相似的样本在模式空间处于相近的位置，如何度量两个样本在模式空间的远近？这里可应用在几何学中的距离的概念，它在模式识别中是作为与“相似性”相反的“相异性”的度量使用的。任何满足下述三个条件的“相异性”均可定义为模式空间的距离：

① $D_{ij} \geqslant 0$；　　如　　$x_i = x_j$　，　$D_{ij} = 0$

② $D_{ij} = D_{ji}$

③ $D_{ik} + D_{jk} \geqslant D_{ij}$

条件①和条件②表明距离应取正值且是对称的，条件③是作为“度量”距离与“非度量”距离的差异。在欧氏空间中，两点间的直线距离恒小于或等于其他路径的长度。

Minkowski 距离：是一种广义距离，对于 n 维空间的两个模式矢量 x_i 与 x_j，定义为

$$D_{ij,\mathrm{Minkowski}} = \{\sum (x_{ik} - x_{jk})^d\}^{1/d} \tag{7-1}$$

特殊情况，当 d 等于 2 时，即得我们常用的欧氏距离(Euclidian 距离)：

$$D_{ij,\mathrm{Euclidian}} = \{\sum (x_{ik} - x_{jk})^2\}^{1/2} \tag{7-2}$$

Manhatten 距离：

$$D_{ij,\mathrm{Manhatten}} = \sum |x_{ik} - x_{jk}| \tag{7-3}$$

Haming 距离：

$$D_{ij,\mathrm{Haming}} = \sum \mathrm{XOR}(x_{ik} - x_{jk}) \tag{7-4}$$

Tanimoto 距离：

$$D_{ij,\mathrm{Tanimoto}} = 1 - [\sum \mathrm{AND}(x_{ik} - x_{jk}) / \sum \mathrm{OR}(x_{ik} - x_{jk})] \tag{7-5}$$

上述式中，AND，OR 和 XOR 分别为逻辑操作的“与”、“或”和“排斥”。所以，Haming 距离和 Tanimoto 距离一般多用于定性数值，即 0，1 类型数据之中。

如果样本是服从多元正态分布的，一般还可采用 Mahalanobis 距离，习惯上称为马氏距离。其定义为：设 x_i，x_j 是从均值为 μ、协方差阵为 V 的母体 G 中的样本，则它们之间的 Mahalanobis 距离为

$$D_{ij,\mathrm{Mahalanobis}} = (x_i - x_j)' V^{-1} (x_i - x_j) \tag{7-6}$$

而 x_i 与母体 G 的距离则为它与均值 μ 的距离，

$$D_{i\mu,\mathrm{Mahalanobis}} = (x_i - \mu)' V^{-1} (x_i - \mu)$$

马氏距离在判别分析中的距离判别中有着重要作用。

除作为“相异性”度量的距离外，还可以定义在此基础上的“相似性”测度。模式矢量 x_i 与 x_j 的“相似性”测度 S_{ij} 可定义为

$$S_{ij} = 1 - [D_{ij,\mathrm{Minkowski}} / \mathrm{Max}(D_{\mathrm{Minkowski}})] \tag{7-7}$$

式中，Max ($D_{\mathrm{Minkowski}}$) 是所有被研究的数据中相距最远的两个模式矢量之间的距离。显然，对于同一模式矢量，S_{ii} 等于 1，即完全相似，实因 D_{ii} 等于零；对于该数据集中相距最远的两个模式矢量，S_{jk} 等于 0，即最不相似，实因 D_{jk} 等于 Max(D)。

另外，常用的还有夹角余弦和相关系数。

夹角余弦

$$\cos(\alpha_{ij}) = (\sum x_{ik} \times x_{jk})/(\sqrt{x_{ik}^2} \times \sqrt{x_{jk}^2})$$

如进行谱图比较时,此夹角余弦是一个很好的相似性度量指标。

相关系数

$$\cos(\alpha_{ij})=[\sum(x_{ik}-m_i)\times(x_{jk}-m_j)]/(\sqrt{(x_{ik}-m_i)^2}\times\sqrt{(x_{jk}-m_j)^2})$$

式中,m_i 和 m_j 分别表示第 i 和第 j 个样本的均值,即 $m_i=(\sum x_{ik})/n, k=1,2,\cdots,n$。

文献 [1] 对各种距离和相似系数作了一个汇总介绍，读者可参阅该文献。

第二节 特征抽取方法

选取合适的特征量，张成合适的特征空间，是模式识别工作成败的一个关键，特征量的选取标准可分为化学（或物理）的和数学的两方面。化学（或物理）的选取标准基于对所处理的信息的本质或主要因子的理解。在这方面，因其依赖具体问题的实际情况，在此无法指出普遍有效的方法。由于对所处理的信息了解不深，而且往往有许多因素之间的相关关系及程度亦不能肯定，因此人们通常是先根据化学（或物理）的选取标准尽可能把一切可能有关、又易于获得的因素（特征量）都提出来，然后借助于数学方法，筛选出对分类作用大的特征量，张成合适的特征空间供分析使用。在实际的计算中，人们总是力图抛弃那些对分类作用不大的特征量，使特征量的数目(在保证良好的分类效果的前提下)减到最小。这是因为：① 多余的特征量不但没有多大好处，而且可能干扰分类过程；② 为了保证样本数和模式空间维数的比值大于 3（最好是大于 10），最好使模式空间维数（特征量数）降至最小。

上述特征抽取的说法似乎与前节中讨论的维数越高，模式识别方法的识别能力越强之说相悖。实际上，在这里有一个特征变量本身的判别能力和它们之间存在的相关问题。可以这样说，如果每个特征变量都具有一定的判别能力且是独立的，这时维数提高使模式识别方法的识别能力将增强之说是正确的，因为每增加一维就增加了新的信息；然而，如果特征变量相互相关，此时，多余的特征变量不但没有多大好处，而且可能干扰分类过程，这是由于增加的特征变量并未增加判别信息，面只是增加了运算量且还可能带来模型不稳定的问题。也正是因为这一点，特征抽取是模式识别工作成败的一个关键。因篇幅所限，在此不进行更多的讨论，读者可参阅文献 [2]。以下介绍两种单变量的特征抽取方法。

设原始的模式矢量含有 n 个参量，即 $\mathbf{x}_i=(x_{i1},x_{i2},\cdots,x_{in})$。在所有的样本中，可以设想，方差较大的参量，即不同样本取值差异较大的参量，对分类应当贡献较大。反之，如某一变量的方差很小（极端的情况下是该变量对所有的样本取相同数值，方差为零），这种变量对于分类而言显然无意义。因此可以变量 i 的方差 V_j 作为权因子，来度量变量 i 的重要性。

$$V_j=\{\sum(x_{ij}-m_j)^2\}/(n-1) \tag{7-8}$$

式中，m_j 为变量 i 的均值

$$m_j=\sum(x_{ij,\text{old}})/n$$

在选择变量时，将 V_j 小的变量优先弃去。

对于训练集包含已知分类的样本，在这里介绍一种简单而又有效的特征抽取的方法，它可以容易辨别出特征变量对分类的贡献大小。这种评估特征变量的标准称为 Fisher 权重

$$F_i=(m_{j1}-m_{j2})^2/(V_{j1}+V_{j2}) \tag{7-9}$$

式中，m_{j1}和 m_{j2}分别是训练集中类 1 和类 2 的 i 参量的均值，V_{j1}与 V_{j2}是这两类样本计算出的 i 变量的方差，权重大，表明该变量 i 对两个类而言差别大，应优先选用。

第三节　模式识别的数据预处理方法

在初步选取特征变量后，模式识别分类计算之前，有几种应该采取的数据预处理方法(data preprocessing)，使模式空间样本代表点分布结构改变，变得更有利于作分类运算。例如，各特征变量取不同量纲的数值时，如果有的变量取的绝对值很大，而另外一些变量的绝对值很小，则在分类过程中不利于获得最佳结果，因此应当用标度化方法使各变量的变化幅度处于同一水平上。下面介绍几种这样的标度化方法。

(1) 范围标度化 (Range scaling)

$$x_{ij,\text{new}}=(x_{ij,\text{old}}-x_{ij,\text{最小}})/(x_{ij,\text{最大}}-x_{ij,\text{最小}}) \tag{7-10}$$

在此，$x_{ij,\text{old}}$代表第 i 个样本的原有的第 j 分量，$x_{ij,\text{new}}$代表第 i 个样本经换算后的第 j 分量，$x_{ij,\text{最大}}$代表原有的第 j 分量的最大值，$x_{ij,\text{最小}}$则代表原有的第 j 分量的最小值。范围标度化的一个缺点是，若数值集合中有一个数值很大，则其余各值都会相差不大。

(2) 自标度化 (Autoscaling)

$$x_{ij,\text{new}}=(x_{ij,\text{old}}-m_j)/V_j \tag{7-11}$$

式中，V_j 为变量 j 的方差，m_j 为变量 j 的均值。自标度法是一个很好的方法，经此变换后，各变量在分类过程中影响程度相同。

(3) 标准化 (Normalization)　标准化预处理的方法很多，对不同的仪器有不同的方法，如在色谱数据中常采用面积归一化，以减小进样误差的影响，即

$$x_{ij,\text{new}}=x_{ij,\text{old}}/\sum(x_{ij,\text{old}}) \tag{7-12}$$

而对于质谱数据则常采用最大归一化预处理的方法，即

$$x_{ij,\text{new}}=x_{ij,\text{old}}/\max(x_{ij,\text{old}}) \tag{7-13}$$

(4) 变换法　另外，在化学模式识别中，还常采用变换方法来对数据进行预处理。常用的有

$$x_{ij,\text{new}}=\sqrt{x_{ij,\text{old}}} \tag{7-14}$$

$$x_{ij,\text{new}}=\lg(x_{ij,\text{old}}) \tag{7-15}$$

以达到改变数据标度的目的。

(5) 组合法　根据不同的情况，将原来的变量，按照一定的方式，如变量相加、变量相减或变量相比等进行组合以产生新的变量。

总之，对于不同的情况可以采用不同的预处理方法，这不但需要正确理解各种预处理方法的物理意义，同时也必须结合不同问题的具体化学经验来对原始数据进行预处理，不可统而论之。

第四节　有监督的模式识别方法——判别分析法

有监督的模式识别的分类方法很多，这类方法的总体基本思路都是用一组已知类别的样本作为训练集，即用已知的样本进行训练，让计算机向这些已知样本“学习”，这种求取分类器的模式识别方法称为“有监督的学习”或“有老师的学习”，这里训练集就是老师。并由这个训练集得到判别模型。这类方法一般可分为参数法和非参数法两类，其中参数法一般由统计学家提出，其判别效果的好坏依赖于样本是否符合假设的统计分布，而非参数判别分析法多由实验科学家或计算机模式识别专家提出，对样本分布没有特殊要求。在这里我们将介绍

几种最常用的方法，包括几种经典的参数方法，如距离判别分析法、Fisher 判别分析法、Beayes 判别分析法、逐步判别分析法，以及几种非参数方法，如线性学习机、K 邻域判别法、势函数判别法、人工神经网络判别法。

为检测所得的判别模型的判别能力，还常用另外一组已知类别的样本组成测试集。在训练中所得的正确判别率称为识别率，而用测试集所得的正确判别率则称为预测率，一般说来，预测率对模型好坏的判别是比识别率更重要些。

一、距离判别法

【基本思路】

判别问题用统计学家的语言来说就是，有 k 个总体 $G_1,G_2,\cdots,G_k$，它们的分布函数分别是 $F_1(\boldsymbol{x}),\cdots,F_k(\boldsymbol{x})$，其中 $\boldsymbol{x}$ 为 n 维的样本矢量。对给定的样本 $\boldsymbol{x}$，判别分析法的目的就是要判决它来自哪个母体。一个最直观的方法就是计算样本 $\boldsymbol{x}$ 到 $G_1,G_2,\cdots,G_k$ 的距离 $\mathrm{d}(\boldsymbol{x},G_1),\mathrm{d}(\boldsymbol{x},G_2),\cdots,\mathrm{d}(\boldsymbol{x},G_k)$，然后根据样本 $\boldsymbol{x}$ 到它们的距离来确定样本的归宿，即

$$\boldsymbol{x}\in G_i,\qquad \text{如 } \mathrm{d}(\boldsymbol{x},G_i)=\min\{\mathrm{d}(\boldsymbol{x},G_1),\mathrm{d}(\boldsymbol{x},G_2),\cdots,\mathrm{d}(\boldsymbol{x},G_k)\}$$

这也就构成了距离判别的基本思路。下面，以两个总体的情况来具体说明。

设有两个正态分布总体 G_1,G_2，它们的均值分别为 $\mu^{(1)}$ 和 $\mu^{(2)}$，协方差阵分别为 $\boldsymbol{V}_1$ 和 $\boldsymbol{V}_2$。对给定的一个样本 $\boldsymbol{x}$，距离判别的方法就是根据它到母体的距离来判决它来自 G_1 还是 G_2，即

$$\begin{cases}\boldsymbol{x}\in G_1, & \mathrm{d}(\boldsymbol{x},G_1)<\mathrm{d}(\boldsymbol{x},G_2)\\ \boldsymbol{x}\in G_2, & \mathrm{d}(\boldsymbol{x},G_1)>\mathrm{d}(\boldsymbol{x},G_2)\end{cases}$$

此时，问题是该采用什么样的距离来定义样本到母体的距离。对于这种具有分布函数的母体，前节讨论的 Mahalanobis 距离比较合适。下面将简要介绍距离判别的基本算法。

【基本算法】

(1) 设有两个正态分布总体 G_1，G_2，它们的均值分别为 $\mu^{(1)}$ 和 $\mu^{(2)}$，协方差阵相等，即都为 $\boldsymbol{V}$ 的情况。

① 计算任意样本 $\boldsymbol{x}$ 到母体 G_1，G_2 的 Mahalanobis 距离。样本 $\boldsymbol{x}$ 到母体 G_1、G_2 的距离实际上是通过样本 $\boldsymbol{x}$ 到该母体的均值矢量 $\mu^{(1)}$ 和 $\mu^{(2)}$ 的 Mahalanobis 距离来实现的，即

$$\mathrm{d}^2(\boldsymbol{x},G_1)=(\boldsymbol{x}-\mu^{(1)})^{\mathrm{t}}\boldsymbol{V}^{-1}(\boldsymbol{x}-\mu^{(1)})$$
$$\mathrm{d}^2(\boldsymbol{x},G_2)=(\boldsymbol{x}-\mu^{(2)})^{\mathrm{t}}\boldsymbol{V}^{-1}(\boldsymbol{x}-\mu^{(2)})$$

② 计算 $\mathrm{d}^2(\boldsymbol{x},G_1)-\mathrm{d}^2(\boldsymbol{x},G_2)$，即

$$\begin{aligned}&\mathrm{d}^2(\boldsymbol{x},G_1)-\mathrm{d}^2(\boldsymbol{x},G_2)\\&=(\boldsymbol{x}-\mu^{(1)})^{\mathrm{t}}\boldsymbol{V}^{-1}(\boldsymbol{x}-\mu^{(1)})-(\boldsymbol{x}-\mu^{(2)})^{\mathrm{t}}\boldsymbol{V}^{-1}(\boldsymbol{x}-\mu^{(2)})\\&=-2[\boldsymbol{x}-(\mu^{(1)}+\mu^{(2)})/2]^{\mathrm{t}}\boldsymbol{V}^{-1}(\mu^{(1)}-\mu^{(2)})\end{aligned}$$

令 $\mathrm{d}^2(\boldsymbol{x},G_1)-\mathrm{d}^2(\boldsymbol{x},G_2)=\omega(\boldsymbol{x})$，$(\mu^{(1)}+\mu^{(2)})/2=\mu$，即有

$$\begin{aligned}\omega(\boldsymbol{x})&=-2[\boldsymbol{x}-(\mu^{(1)}+\mu^{(2)})/2]^{\mathrm{t}}\boldsymbol{V}^{-1}(\mu^{(1)}-\mu^{(2)})\\&=-2(\boldsymbol{x}-\mu)^{\mathrm{t}}\boldsymbol{V}^{-1}(\mu^{(1)}-\mu^{(2)})\end{aligned}$$

③ 建立相应判别准则，即

$$\begin{cases}\boldsymbol{x}\in G_1, & \omega(\boldsymbol{x})<0\\ \boldsymbol{x}\in G_2 & \omega(\boldsymbol{x})>0\end{cases}$$

对于母体协方差阵相等的情况，距离判别函数是线性的。

(2) 设有两个正态分布总体 G_1，G_2，它们的均值分别为 $\mu^{(1)}$和 $\mu^{(2)}$，协方差阵不等，即分别都为 $\boldsymbol{V}_1$ 和 $\boldsymbol{V}_2$ 的情况。

① 计算任意样本 $\boldsymbol{x}$ 到母体 G_1，G_2 的 Mahalanobis 距离：

$$d^2(\boldsymbol{x},G_1)=(\boldsymbol{x}-\mu^{(1)})^t\boldsymbol{V}_1^{-1}(\boldsymbol{x}-\mu^{(1)})$$
$$d^2(\boldsymbol{x},G_2)=(\boldsymbol{x}-\mu^{(2)})^t\boldsymbol{V}_2^{-1}(\boldsymbol{x}-\mu^{(2)})$$

② 建立相应判别准则，即

$$\begin{cases}\boldsymbol{x}\in G_1, & d^2(\boldsymbol{x},G_1)<d^2(\boldsymbol{x},G_2)\\ \boldsymbol{x}\in G_2, & d^2(\boldsymbol{x},G_1)>d^2(\boldsymbol{x},G_2)\end{cases}$$

对于母体协方差阵不等的情况，判别函数将是非线性的。

上述算法很容易拓广至多个母体判别。

二、Bayes 判别法

Bayes 判别法的主要思路是建立在条件概率和统计判决理论的 Bayes 定理之上（参见第十章）。在介绍 Bayes 判别法之前，先给出一个有关空间划分的定义。

定义　令 D_1，D_2，…，D_k 是 n 维实数空间的子集，如果它们互不相交，且它们的和就是整个 n 维实数空间，则称 D_1，D_2，…，D_k 是此 n 维实数空间的一个划分。前节讨论的距离判别法，实质上也是建立了这样的一种划分。

【基本思路】

设有 k 个母体 G_1，G_2，…，G_k，它们的分布密度函数分别是 $f_1(x)$，…，$f_k(\boldsymbol{x})$（不一定是正态分布），其中 $\boldsymbol{x}$ 为 n 维的样本矢量。另外，还假定这 k 个母体出现的验前概率为 q_1，q_2，…，q_k，并 $L(i,j)$ 表示样本 $\boldsymbol{x}$ 来自 G_i 而误判为 G_j 的损失，则对给定的一个样本 $\boldsymbol{x}$，Bayes 判别方法的核心就是找到一个划分，使得在此划分的基础上计算样本 x 被误判的平均损失，即使得由（7-16）式表出的平均损失达到最小：

$$g(D_1,D_2,\cdots,D_k)=\sum_{i=1}^{k}q_i\sum L(i,j)p(j|i,D_1,D_2,\cdots,D_k) \tag{7-16}$$

式中，$p(j|i,D_1,D_2,\cdots,D_k)$表示样本 $\boldsymbol{x}$ 来自 G_i 而误判为 G_j 的概率，即

$$p(j|i,D_1,D_2,\cdots,D_k)=\int_{D_j}f_i(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{7-17}$$

它表示 $f_i(\boldsymbol{x})$在划分 D_j 上的积分，也即来自 G_i 的样本 $\boldsymbol{x}$ 出现于划分 D_j 中的误判概率。下面以 $k=2$，即两类判别情况来加以具体说明。

【基本算法】

Bayes 判别可通过以下步骤来实现：

① 根据经验确定两个母体出现的验前概率为 q_1,q_2；

② 根据判别要求确定误判损失函数 $L(i,j)$值，$(i=1,2;j=1,2)$如无特殊要求，一般设

$$L(i,j)=\delta_{ij}$$

即，如 $\boldsymbol{x}\in G_2,L(1,2)=0,\boldsymbol{x}\in G_2,L(1,2)=1$。

③ 计算 Bayes 判别函数 $\omega_i(\boldsymbol{x})(i=1,2)$：

$$\omega_1(\boldsymbol{x})=q_1L(1,2)f_1(\boldsymbol{x})$$
$$\omega_2(\boldsymbol{x})=q_2L(2,1)f_2(\boldsymbol{x}) \tag{7-18}$$

如果 $\omega_1(\boldsymbol{x})<\omega_2(\boldsymbol{x})$，则样本 $\boldsymbol{x}$ 属于 G_1，反之，样本 $\boldsymbol{x}$ 属于 G_2。由这样的判别函数，可以得到一个划分 D_1、D_2。

在上述 Bayes 判别中，因存在一个分布密度函数 $f_i(\boldsymbol{x})$，它是不确定的，一般都假设每一类的样本服从多元正态分布，所以，在一般介绍 Bayes 判别的书和文献中，都是以多元正态分布密度函数代入（7-18）式来进行计算的，即

$$f_1(\boldsymbol{x})=\{1/[(2\pi)^{n/2}|\boldsymbol{V}_1|^{1/2}]\}\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})'\boldsymbol{V}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})]/2\} \tag{7-19a}$$

$$f_2(\boldsymbol{x})=\{1/[(2\pi)^{n/2}|\boldsymbol{V}_2|^{1/2}]\}\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})'\boldsymbol{V}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})]/2\} \tag{7-19b}$$

式中，$\boldsymbol{V}_1$，$\boldsymbol{V}_2$ 分别表示第一及第二类的协方差阵，$\boldsymbol{\mu}^{(1)}$，$\boldsymbol{\mu}^{(2)}$ 分别表示第一及第二类的均值向量。这样，前述的 Bayes 判别函数 $\omega_i(\boldsymbol{x})(i=1,2)$ 可表述为

$$\omega_1(\boldsymbol{x})=q_1L(1,2)\{1/[(2\pi)^{n/2}|\boldsymbol{V}_1|^{1/2}]\}\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})'\boldsymbol{V}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})]/2\} \tag{7-20a}$$

$$\omega_2(\boldsymbol{x})=q_2L(2,1)\{1/[(2\pi)^{n/2}|\boldsymbol{V}_2|^{1/2}]\}\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})'\boldsymbol{V}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})]/2\} \tag{7-20b}$$

又因为 $L(i,j)=\delta_{ij}$，所以有 $L(1,2)$ 和 $L(2,1)$ 都为 1，故(7-20)式变为

$$\omega_1(\boldsymbol{x})=q_1\{1/[(2\pi)^{n/2}|\boldsymbol{V}_1|^{1/2}]\}\exp\{-(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})'\boldsymbol{V}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})\} \tag{7-21a}$$

$$\omega_2(\boldsymbol{x})=q_2\{1/[(2\pi)^{n/2}|\boldsymbol{V}_2|^{1/2}]\}\exp\{-(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})'\boldsymbol{V}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})\} \tag{7-21b}$$

如果 $\omega_1(\boldsymbol{x})<\omega_2(\boldsymbol{x})$，即

$$q_1\{|\boldsymbol{V}_1|^{1/2}\}\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})'\boldsymbol{V}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})]/2\}<$$
$$q_2\{|\boldsymbol{V}_2|^{1/2}\}\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})'\boldsymbol{V}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})]/2\}$$

或

$$q_1/q_2<\{|\boldsymbol{V}_2|^{1/2}\}\exp\{-(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})'\boldsymbol{V}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})\}/$$
$$\{|\boldsymbol{V}_1|^{1/2}\}\exp\{-(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})'\boldsymbol{V}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})\}$$

则样本 $\boldsymbol{x}$ 属于 G_1。一般说来，q_1/q_2 可看成常数，如设为 c，则上式可变为

$$\{|\boldsymbol{V}_2|^{1/2}\}\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})'\boldsymbol{V}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})]/2\}/$$
$$\{|\boldsymbol{V}_1|^{1/2}\}\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})'\boldsymbol{V}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})]/2\}>c \tag{7-22}$$

如再对(7-22)式加一假设，即设第一及第二类的协方差阵为等协方差阵，此时有

$$\boldsymbol{V}_2=\boldsymbol{V}_1=\boldsymbol{V}$$

则(7-22)式可简化为

$$\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})'\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})]/2\}/\exp\{-[(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})'\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})]/2\}>c \tag{7-23}$$

亦即

$$\exp\{-(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})'\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})+(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})'\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})\}>c \tag{7-24}$$

对(7-24)式两边取对数

$$\{-(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})'\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(2)})+(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})'\boldsymbol{V}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^{(1)})\}>\log(c) \tag{7-25}$$

亦即

$$[\boldsymbol{x}-(\boldsymbol{\mu}^{(1)}+\boldsymbol{\mu}^{(2)})/2]'\boldsymbol{V}^{-1}(\boldsymbol{\mu}^{(1)}-\boldsymbol{\mu}^{(2)})>\log(c)$$

当两类样本的验前概率相等时，即此时有 $q_1=q_2$，$\log(c)=0$。此时 Bayes 判别与前节讨论的距离判别完全一致了。这说明当两母体验前概率相等，且误判损失函数为常数时，距离判别与 Bayes 判别是完全等价的。

三、Fisher 判别法

Fisher 判别法是借助方差分析的思想来导出判别函数的。它力争找到一个最佳线性投影

方向，使得沿此方向投影，可使类间方差与类内方差的比值达到最大，故 Fisher 判别法一般是采用线性判别函数来进行判别的。

【基本思路】

Fisher 判别法的基本思路是先计算出不同类别样本的组间方差和组内方差，而后构造一个组间方差和组内方差的比值。显然，如能找到一个分类方法，可使得组间方差和组内方差的比值达到最大，当然也就找到了一个最佳的判别方法。

设有 k 个母体 G_1，G_2，…，G_k，它们的相应均值和协方差阵分别为分布密度函数，分别是 $\mu^{(1)},\mu^{(2)},\cdots,\mu^{(k)};\boldsymbol{V}_1,\boldsymbol{V}_2,\cdots\boldsymbol{V}_k$。任给一个样本矢量 $\boldsymbol{x}$，它的一个线性函数由 $\boldsymbol{u}(\boldsymbol{x})=\boldsymbol{u}'\boldsymbol{x}$ 表示出，则母体 G_i 的均值和方差的线性函数可表为

$$e_i=\mathrm{E}[\boldsymbol{u}(\boldsymbol{x})|G_i]=\boldsymbol{u}'\mu^{(i)} \qquad i=1,2,\cdots,k$$

$$v_i^2=\mathrm{Var}[\boldsymbol{u}(\boldsymbol{x})|G_i]=\boldsymbol{u}'\boldsymbol{V}_i\boldsymbol{u} \qquad i=1,2,\cdots,k$$

记

$$B_0=\sum_{i=1}^{k}[e_i-\sum_{i=1}^{k}(e_i)/k]^2 \tag{7-26}$$

$$E_0=\sum_{i=1}^{k}v_i^2=\sum_{i=1}^{k}\boldsymbol{u}'\boldsymbol{V}_i\boldsymbol{u} \tag{7-27}$$

因 e_i 和 v_i^2 都是标量，所以 B_0 就相当于一元方差分析中的组间方差总和，E_0 就相当于组内方差的总和。如选择一个 $\boldsymbol{u}$ 使

$$\Delta(\boldsymbol{u})=B_0/E_0 \tag{7-28}$$

达到极大，这个 $\boldsymbol{u}$ 就是我们要寻找的最佳投影方向。令

$$\boldsymbol{\Theta}'=[\mu^{(1)},\mu^{(2)},\cdots,\mu^{(k)}]=\begin{bmatrix}\mu_1^{(1)} & \mu_1^{(2)} & \cdots & \mu_1^{(k)}\\ \mu_2^{(1)} & \mu_2^{(2)} & \cdots & \mu_2^{(k)}\\ \vdots & \vdots & & \vdots\\ \mu_n^{(1)} & \mu_n^{(2)} & \cdots & \mu_n^{(k)}\end{bmatrix}$$

则

$$B=\boldsymbol{\Theta}'[I-(1/k)J]\boldsymbol{\Theta} \tag{7-29}$$

式中，$\boldsymbol{I}$ 为 k 阶单位矩阵，$\boldsymbol{J}$ 为 k 阶纯 1 矩阵，即

$$\boldsymbol{J}=[1,1,\cdots,1]=\begin{bmatrix}1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1\\ \vdots & \vdots & & \vdots\\ 1 & 1 & \cdots & 1\end{bmatrix}$$

此时，有

$$\begin{aligned}B_0&=\boldsymbol{u}'\boldsymbol{B}\boldsymbol{u}=\boldsymbol{u}'\boldsymbol{\Theta}'[\boldsymbol{I}-(1/k)\boldsymbol{J}]\boldsymbol{\Theta}\boldsymbol{u}\\ &=\boldsymbol{u}'\boldsymbol{\Theta}'\boldsymbol{\Theta}\boldsymbol{u}-(1/k)\boldsymbol{u}'\boldsymbol{\Theta}'\boldsymbol{J}\boldsymbol{\Theta}\boldsymbol{u}\\ &=[\boldsymbol{u}'\mu^{(1)},\boldsymbol{u}'\mu^{(2)},\cdots,\boldsymbol{u}'\mu^{(k)}]\begin{bmatrix}\mu^{(1)'}\boldsymbol{u}\\ \mu^{(2)'}\boldsymbol{u}\\ \vdots\\ \mu^{(k)'}\boldsymbol{u}\end{bmatrix}-(1/k)\boldsymbol{u}'\boldsymbol{\Theta}'\boldsymbol{J}\boldsymbol{\Theta}\boldsymbol{u}\\ &=\sum_{i=1}^{k}[e_i-\sum_{i=1}^{k}(e_i)/k]^2\end{aligned}$$

在上述计算时，利用方差平方和的分解性质，即其平方和的交叉项等于零的性质。另外，令

$$E=\sum_{i=1}^{k}V_i$$

则

$$E_0=\sum_{i=1}^{k}v_i^2=\sum_{i=1}^{k}u^tV_iu=u^tEu \tag{7-30}$$

于是
$$\Delta(u)=B_0/E_0=(u^tBu)/(u^tEu)$$

对 $\Delta(u)$求导，即求 $\partial[\Delta(u)]/\partial u=0$，此时，有

$$|B-\lambda E|=0 \tag{7-31}$$

亦即 $\Delta(u)$的极大值为上述方程的最大特征根，而此最大特征根所对应的特征向量即为最佳投影方向。下将以两类判别情况来说明 Fisher 判别法的基本算法，拓广至多类判别是显然的。

【基本算法】

对于两类来自不同母体的样本，$x_1^{(1)},x_2^{(1)},x_3^{(1)},\cdots,x_{n1}^{(1)}$ 和 $x_1^{(2)},x_2^{(2)},x_3^{(2)},\cdots,x_{n2}^{(2)}$，其中 $x_j^{(i)}(i=1,2;j=1,2,\cdots,n_1$ 或 $n_2)$表示第 i 类第 j 个样本，$\mu^{(1)},\mu^{(2)};V_1,V_2$ 分别表示此两类样本的均值向量和协方差阵。Fisher 判别可通过以下步骤来实现：

(1) 计算此两个母体样本的类间方差 B 与类内方差 E

$$B=(1/2)(\mu^{(1)}-\mu^{(2)})(\mu^{(1)}-\mu^{(2)})^t$$
$$E=V_1+V_2$$

则此时有

$$\Delta(u)=(u^tBu)/(u^tEu)=[(1/2)u^t(\mu^{(1)}-\mu^{(2)})(\mu^{(1)}-\mu^{(2)})^tu]/[u^t(V_1+V_2)u]$$

(2) 计算 $\partial[\Delta(u)]/\partial u$，并令其等于零，可得

$$(V_1+V_2)u=\mu^{(1)}-\mu^{(2)} \tag{7-32}$$

即

$$u=(V_1+V_2)^{-1}(\mu^{(1)}-\mu^{(2)}) \tag{7-33}$$

(3) 计算 Fisher 判别函数 $\omega(x)$

$$\omega(x)=u^t(x-\mu)$$

式中，$\mu=(1/2)(\mu^{(1)}-\mu^{(2)})$，如果 $\omega(x)<0$，则样本 x 属于 G_1，反之，样本 x 属于 G_2。由这样的判别函数，可以得到一个划分 D_1,D_2。

$$D_1=\{x: u^t(x-\mu)>0\},\qquad D_2=\{x: u^t(x-\mu)\leqslant 0\}$$

在此是从母体分布并假设它们的均值向量和协方差阵已知的条件来讨论的，而在实际的应用中，这些参数如均值向量和协方差阵都是通过样本估计而得，如设样本均值向量和样本协方差阵分别为 $\bar{x}_i$ 和 $S_i(i=1,2)$，则有

$$V_1=[1/(n_1-1)]S_1 \tag{7-34a}$$
$$V_2=[1/(n_2-1)]S_2 \tag{7-34b}$$

所以

$$(V_1+V_2)=[1/(n_1-1)]S_1+[1/(n_2-1)]S_2 \tag{7-35}$$

将(7-35)式代入(7-33)式，另外，再以 $\bar{x}_1$ 和 $\bar{x}_2$ 代替 $\mu^{(1)},\mu^{(2)}$，即可得到实际样本的判别函数。

实际上，Fisher 判别法的计算很简单，即只需计算出相应不同类别样本的均值向量和协方差阵，就可容易得到 Fisher 判别函数了。

四、逐步判别分析法

在实际的判别分析中，人们总是力图抛弃那些相互相关而对分类作用不大的特征变量，使特征变量的数目（在保证良好的分类效果的前提下）减到最小。这是因为：① 多余的特征变量不但没有多大好处，而且可能干扰分类过程；② 为了保证样本数和模式空间维数的比值大于 3（最好是大于 10），最好使模式空间维数（特征量数）降至最小。本节要讨论的逐步判别分析法就是这样的一种方法。

为检验所含特征变量对判别效果显著性，有必要在此先介绍关于判别检验的两个统计量及其检验方法。

（一）T^2统计量

T^2 统计量一般可用来检验两类样本之间的判别效果。T^2 统计量本质上是用来进行均值向量的检验的，它可用于检验两个均值向量是否具有显著性差异，即检验是否

$$\overline{x}^{(1)}=\overline{x}^{(2)}$$

试想如果两个均值向量不具有显著性差异，则很难认为它们是来自两个不同母体，反之，如果两个均值向量确实存在显著性差异，则可肯定它们不是来自一个母体。用 T^2 统计量来检验两类样本所包含的各个变量的判别效果就基于上述事实。

$$T^2=[n_1 n_2/(n_1+n_2)](\overline{x}_1-\overline{x}_2)'S^{-1}(\overline{x}_1-\overline{x}_2) \tag{7-36}$$

式中，$\overline{x}_1$ 和 $\overline{x}_2$ 分别为两类样本的样本均值向量，n_1、n_2 分别为第一类和第二类样本的个数

$$S=[1/(n_1+n_2-2)]\sum_{i=1}^{2}\sum_{k=1}^{n_i}(x_{ik}-\overline{x})(x_{ik}-\overline{x})'$$

由于，$F=\{(n_1+n_2-n-1)/[(n_1+n_2-2)n]\}T^2$

遵从自由度为 n 和(n_1+n_2-n-1)的 $F(n,n_1+n_2-n-1)$分布。其中 n 是样本向量所含变量数，即其维数。当 $F>F_\alpha$，则表明两个母体的均值有显著性差异，亦即样本的 n 个变量可以区分此两个母体，对它们的判别效果显著。

（二）Wilks 统计量

Wilks 统计量是用来检验多类样本的估计均值向量是否存在显著性差异的，所以它可用于多母体之间的判别效果。即检验是否

$$\overline{x}^{(1)}=\overline{x}^{(2)}=\overline{x}^{(3)}=\cdots=\overline{x}^{(k)}$$

设从 k 个母体 $G_1,G_2,\cdots,G_k$ 分别得到如下样本，

$$\begin{matrix} x_1^{(1)}, & x_2^{(1)}, & x_3^{(1)}, & \cdots, & x_{n1}^{(1)} \\ x_1^{(2)}, & x_2^{(2)}, & x_3^{(2)}, & \cdots, & x_{n2}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ x_1^{(k)}, & x_2^{(k)}, & x_3^{(k)}, & \cdots, & x_{nk}^{(k)} \end{matrix}$$

其中，$x_j^{(i)}(i=1,2,\cdots,k;j=n_1,n_2,\cdots,n_k)$表示第 i 类第 j 个样本，它们都是 m 维向量，即含有 m 个变量的向量，各个母体的相应均值向量和协方差阵分别为 $\mu^{(1)},\mu^{(2)},\cdots,\mu^{(k)};V_1,V_2,\cdots,V_k$。令

$$n=\sum_{i=1}^{k}n_i;\ X_\alpha=[x_1^{(\alpha)}x_2^{(\alpha)}x_3^{(\alpha)}\cdots x_{n_\alpha}^{(\alpha)}]\quad \alpha=1,2,\cdots,k$$

$$X=[X_1'X_2'\cdots X_k']'$$

则总体的估计均值 $\overline{x}$及每类的估计均值 $\overline{x}^{(\alpha)}$分别为

$$\bar{x}=(1/n)\sum_{\alpha=1}^{k}\sum_{i=1}^{n_\alpha}x_i^{(\alpha)}$$

$$\bar{x}^{(\alpha)}=(1/n_\alpha)\sum_{i=1}^{n_\alpha}x_i^{(\alpha)} \qquad \alpha=1,2,\cdots,k$$

同时令估计总方差阵为 W,即

$$W=(1/n)\sum_{\alpha=1}^{k}\sum_{i=1}^{n_\alpha}(x_i^{(\alpha)}-\bar{x})(x_i^{(\alpha)}-\bar{x})^t=X^t(I_n-(1/n)J)X \tag{7-37}$$

式中,I_n 为 n 阶单位矩阵,J 为 n 阶纯 1 矩阵。将其进行平方和分解,可得

$$W=\sum_{\alpha=1}^{k}\sum_{i=1}^{n_\alpha}(x_i^{(\alpha)}-\bar{x})(x_i^{(\alpha)}-\bar{x})^t \tag{7-38}$$

$$=\sum_{\alpha=1}^{k}\sum_{i=1}^{n_\alpha}(x_i^{(\alpha)}-\bar{x}^{(\alpha)})(x_i^{(\alpha)}-\bar{x}^{(\alpha)})^t+\sum_{\alpha=1}^{k}n_\alpha(\bar{x}^{(\alpha)}-\bar{x})(\bar{x}^{(\alpha)}-\bar{x})^t$$

即(7-38)式中的第一项为类内方差,记为 E,而第二项表示类间方差,记为 B,则

$$W=E+B$$

而在多元方差分析中著名的 Wilks 统计量可定义为

$$\Lambda=|E|/|E+B|=|E|/|W| \tag{7-39}$$

式中,$|\cdot|$表示行列式值。其自由度参数分别为 $m,n-k,k-1$,服从 Wilks 分布,记为 $\Lambda(m,n-k,k-1)$。

(三)附加信息检验

所谓附加信息检验就是对另增加的一些变量进行检验,检验它们是否能提供新的信息,亦称附加信息。设从 k 个母体 $G_1, G_2, \cdots, G_k$ 分别得到如下样本.

$$\begin{matrix} x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \cdots, x_{n_1}^{(1)} \\ x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, \cdots, x_{n_2}^{(2)} \\ \vdots \quad \vdots \quad \vdots \quad\quad \vdots \\ x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \cdots, x_{n_k}^{(k)} \end{matrix}$$

式中,$x_j^{(i)}(i=1,2,\cdots,k;j=n_1,n_2,\cdots,n_k)$表示第 i 类第 j 个样本,它们都是 m 维向量,即含有 m 个变量的向量,如通过某种步骤已选中了 r 个变量,要检验另外$(m-r)$个指标对区分母体是否提供附加信息,为了书写方便,不妨设选中是前 r 个变量,记

$$x_i^{(\alpha)}=\begin{bmatrix} x_{i1}^{(\alpha)} \\ x_{i2}^{(\alpha)} \end{bmatrix}\begin{matrix} r \\ m-r \end{matrix} \qquad \begin{matrix} \alpha=1,2\cdots,k \\ I=1,2\cdots,n_\alpha \end{matrix} \tag{7-40}$$

样本的均值和协方差阵也作相应部分

$$\bar{x}^{(\alpha)}=\begin{bmatrix} \bar{x}_1^{(\alpha)} \\ \bar{x}_2^{(\alpha)} \end{bmatrix}\begin{matrix} r \\ m-r \end{matrix} \qquad \alpha=1,2\cdots,k$$

$$E=\begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \quad B=\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

$$W=B+E=\begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}$$

为了检验后$(m-r)$个变量是否提供附加信息，不能简单检验

$$\bar{x}_2^{(1)}=\bar{x}_2^{(2)}=\bar{x}_2^{(3)}=\cdots=\bar{x}_2^{(k)}$$

因为这么做是将前r个变量与后$(m-r)$个变量完全孤立起来，而没有考虑到它们之间的相互作用，比较合理的做法是从后$(m-r)$个变量中将前r个变量与它们的线性相关部分扣除，继续检验剩下部分的均值是否相等，也就是说在给定前r个变量的条件下，来检验后$(m-r)$个变量的条件均值是否相等。

记$\bar{x}_{2\cdot1}^{(a)}$为在给定前r个变量的条件下后$(m-r)$个变量的条件均值，则要检验的假设是

$$\bar{x}_{2\cdot1}^{(1)}=\bar{x}_{2\cdot1}^{(2)}=\cdots=\bar{x}_{2\cdot1}^{(a)}$$

上述检验可通过下述统计量来进行[3]，

$$\Lambda_{(m-r)\cdot(r)}=\frac{|E_{22}-E_{21}E_{11}^{-1}E_{12}|}{|W_{22}-W_{21}W_{11}^{-1}W_{12}|}$$

又因为

$$E=|E_{11}||E_{22}-E_{21}E_{11}^{-1}E_{12}|$$

$$W=|W_{11}||W_{22}-W_{21}W_{11}^{-1}W_{12}|$$

故

$$\Lambda_{(m)}=\frac{|E|}{|W|}=\left(\frac{|E_{11}|}{|W_{11}|}\right)\left(\frac{|E_{22}-E_{21}E_{11}^{-1}E_{12}|}{|W_{22}-W_{21}W_{11}^{-1}W_{12}|}\right)$$

记

$$\Lambda_{(r)}=|E_{11}|/|W_{11}|$$

它是检验前r个变量构成的样本的均值向量是否有显著性差异的 Wilks 统计量，即检验假设

$$\bar{x}_1^{(1)}=\bar{x}_1^{(2)}=\bar{x}_1^{(3)}=\cdots=\bar{x}_1^{(a)}$$

的统计量，所以有

$$\Lambda_{(m)}=\Lambda_{(r)}\Lambda_{(m-r)\cdot(r)} \tag{7-41}$$

(7-41)式很像条件概率的公式，由于 Wilks 统计量有这个好的性质，在逐步判别中使用它将是非常方便的。

另外，Wilks 统计量$\Lambda_{(m)}$还有如下的递推公式：

$$\Lambda_{(m)}=\Lambda_1\Lambda_{2\cdot1}\Lambda_{3\cdot1,2}\cdots\Lambda_{m\cdot1,2,\cdots,(m-1)} \tag{7-42}$$

如记$E=(e_{ij})$，$W=(w_{ij})$，则(6-42)式还可写成

$$\Lambda_{(m)}=(e_{11}/w_{11})(e_{22\cdot1}/w_{22\cdot1})\cdots(e_{mm\cdot1,2,\cdots,(m-1)}/w_{mm.1,2,\cdots,(m-1)}) \tag{7-43}$$

这在逐步判别中使用起来更为方便。

【基本思路】

逐步判别法的基本思路与逐步回归基本是相同的，每一步选取一个判别能力显著的特征变量进入判别函数，然后，对已选出的特征变量逐个进行检验，检验它们是否对正确判别有显著性意义，如其中有某个变量由于新变量的引入而变为不重要、即对正确判别无显著性意义，则将其剔除，直到判别函数中所包含的所有变量判别能力都显著时为止。

本节下面将讨论的逐步判别法，文献［4］对其已作出了详细介绍并附有程序。

【基本算法】

(1) 设在样本$x_i^{(a)}$($i=1,2,\cdots,n_a$;$a=1,2,\cdots,k$)含有m个变量，先从中选出一个变量，计算它的 Wilks 统计量

$$\Lambda_{(i)} = e_{ii}/w_{ii} \qquad i=1,2,\cdots,m$$

使之为此 m 个变量中的最小。为叙述方便又不失一般性，设所挑选的变量是按应选的次序来进行的，即第 r 步正好选中变量 r。第一步选中变量 1，则

$$\Lambda_{(1)} = \min_{1\leqslant i\leqslant m}\Lambda_{(i)} = (e_{ii}/w_{ii}) \qquad i=1,2,\cdots,m$$

考察 $\Lambda_{(1)}$是否落入接受域，即该变量是否对判别有显著性意义，如不显著，则表明一个变量也未选中，不能用判别分析来区分它们；如显著，则进入下一步。

（2）在未选中的变量中，计算它们与已选中的变量 1 配合的 Λ 值

$$\Lambda_{(1i)} = \begin{vmatrix} e_{11} & e_{12} \\ e_{11} & e_{12} \end{vmatrix} \Big/ \begin{vmatrix} w_{11} & w_{12} \\ w_{11} & w_{12} \end{vmatrix} \quad , \quad 2\leqslant i\leqslant m$$

选择 $\Lambda_{(1i)}$达到极小的作为第二个变量，注意到因 $\Lambda_{(1i)} = \Lambda_{(1)}\Lambda_{i\cdot(1)}$，使 $\Lambda_{(1i)}$达到极小等价于使 $\Lambda_{i\cdot(1)}$达到极小。

仿此，如已选入了 r 个变量，不妨设为前 r 个变量，则在未选中的变量中逐次选一个与它们配合，计算

$$\Lambda_{(12\cdots rl)} = \begin{vmatrix} e_{11} & \cdots & e_{1r} & e_{1l} \\ e_{21} & \cdots & e_{2r} & e_{1l} \\ \vdots & & \vdots & \vdots \\ e_{r1} & \cdots & e_{rr} & e_{rl} \\ e_{l1} & \cdots & e_{lr} & e_{rl} \end{vmatrix} \Bigg/ \begin{vmatrix} w_{11} & \cdots & w_{1r} & w_{1l} \\ w_{21} & \cdots & w_{2r} & w_{1l} \\ \vdots & & \vdots & \vdots \\ w_{r1} & \cdots & w_{lr} & w_{rl} \\ w_{l1} & \cdots & w_{lr} & w_{ll} \end{vmatrix} \quad , \quad r\leqslant l\leqslant m$$

选择 $\Lambda_{(12\cdots rl)}$达到极小的作为第 $r+1$ 个变量，由于 $\Lambda_{(12\cdots rl)} = \Lambda_{((12\cdots r)}\Lambda_{l\cdot(12\cdots r)}$，使 $\Lambda_{(12\cdots rl)}$达到极小等价于使 $\Lambda_{l\cdot(12\cdots r)}$达到极小。在此基础上，继续检验新选的第 $r+1$ 个变量能否提供附加信息，如不能则转入(4)，否则转入(3)。

（3）在已选取的 r 个变量中，要考虑到较早选中的变量中其重要性有没有较大的变化，应及时把不能提供附加信息的变量剔除。剔除的原则同于引进的原则，如在已进入的 r 个变量中考察变量 $j(1\leqslant j\leqslant r)$是否需要剔除，就是要计算 $\Lambda_{j\cdot(12\cdot j-1,j+1\cdots r)}$，以选择达到极小的 j，并检验是否显著，如不显著则将该变量剔除，仍进行(3)，继续考察余下的变量是否需要剔除，如显著则回到(2)。

（4）这时既不能选进新变量，又不能剔除已选进的变量，将已选中的变量建立判别函数，前述的 Fisher 或 Bayes 判别函数都可以使用。

从上述算法可以看出，因为每步都要计算一些行列式，似有必要建立一套既经济又方便的计算方法，在此将介绍一种在逐步回归中常用的消去变换法。

如记 $\boldsymbol{E}^{(0)}$为没被变换的类内方差矩阵，对它进行一次消去变换后的矩阵记作 $\boldsymbol{E}^{(1)} = (e_{ij}^{(1)})$，再作一次消去变换后的矩阵记作 $\boldsymbol{E}^{(2)} = (e_{ij}^{(2)})$，…，对类间方差矩阵也有类似的记号。另外，如记第 $r+1$ 次的消去变换为 T_{kk}（即进行以处于第 k 行第 k 列的元素为变换中心元素的消去变换，要求 e_{kk}, w_{kk}不为零，对于所讨论类内方差矩阵和类间方差矩阵，因它们均为正定矩阵，此条件自然满足），则有

$$T_{kk}(\boldsymbol{E}^{(r)}) = \boldsymbol{E}^{(r-1)} = (e_{ij}^{(r+1)}) = \begin{cases} e_{kj}^{(r)}/e_{kk}^{(r)} & j\neq k \\ e_{ik}^{(r)}/e_{kk}^{(r)} & i\neq k \\ 1/e_{kk}^{(r)} & i=j=k \\ e_{ij}^{(r)} - e_{ik}^{(r)}e_{kj}^{(r)}/e_{kk}^{(r)} & i\neq k, j\neq k \end{cases} \tag{7-44a}$$

$$T_{kk}W^{(r)}=W^{(r+1)}=(w_{ij}^{(r+1)})=\begin{cases}w_{kj}^{(r)}/w_{kk}^{(r)} & j\neq k\\ w_{ik}^{(r)}/w_{kk}^{(r)} & i\neq k\\ 1/w_{kk}^{(r)} & i=j=k\\ w_{ij}^{(r)}-w_{ik}^{(r)}w_{kj}^{(r)}/w_{kk}^{(r)} & i\neq k,j\neq k\end{cases} \tag{7-44b}$$

上述消去变换法具有下述三条十分有用的性质，而使之在逐步回归或逐步判别分析中使用起来特别方便。

① $T_{ij}(T_{ij}(\boldsymbol{X}))=\boldsymbol{X}$　即对 $\boldsymbol{X}$ 连续施行两次相同的消去变换，其结果是 $\boldsymbol{X}$ 不变，这很容易直接验证。

② 若 $i\neq k,j\neq l$，则 $T_{ij}(T_{kl}(\boldsymbol{X}))=T_{kl}(T_{ij}(\boldsymbol{X}))$　这表明消去变换在某种意义下的可交换性。此性质也很容易直接验证。

③ 消去变换过程中的子逆矩阵和子行列式计算性质：如记 $\boldsymbol{E}_r$ 为 r 阶正定矩阵，因为 $\boldsymbol{E}^{(1)}=T_{11}(\boldsymbol{E}^{(0)}),\boldsymbol{E}^{(2)}=T_{22}(\boldsymbol{E}^{(2)}),\cdots,\boldsymbol{E}^{(r)}=T_{rr}(\boldsymbol{E}^{(r-1)})$，则 $\boldsymbol{E}_r$ 的行列式值为

$$|\boldsymbol{E}_r|=\prod_{j=1}^{r}e_{jj}^{(j-1)}$$

$$\boldsymbol{E}_r^{-1}=\boldsymbol{E}^{(r)}$$

表 7-1 给出了更清楚的结果。

表 7-1　消去变换过程中的子逆阵和子行列式计算结果

步数	逆矩阵	行列式值	所含变量
1	$e_{11}^{(1)}$	e_{11}	1
2	$e_{11}^{(2)}\quad e_{12}^{(2)}$ $e_{21}^{(2)}\quad e_{22}^{(1)}$	$e_{11}e_{22}^{(1)}$	1,2
	$\vdots\quad\vdots$		
l	$e_{11}^{(l)}\cdots e_{1l}^{(l)}$ $\vdots\quad\vdots$ $e_{l1}^{(l)}\cdots e_{ll}^{(l)}$	$e_{11}e_{22}^{(1)}\cdots e_{ll}^{(l-1)}$	$1,2,\cdots,l$
$r=l+1$	$e_{11}^{(l+1)}\cdots e_{1l}^{(l+1)}e_{1r}^{(l+1)}$ $\vdots\quad\vdots$ $e_{l1}^{(l+1)}\cdots e_{ll}^{(l+1)}e_{1r}^{(l+1)}$ $e_{r1}^{(l+1)}\cdots e_{rl}^{(l+1)}e_{rr}^{(l+1)}$	$e_{11}e_{22}^{(1)}\cdots e_{ll}^{(l-1)}e_{rr}^{(l)}$	$1,2,3,\cdots,l,r$

同时，在上述讨论的消去变换过程中的子逆矩阵和子行列式计算性质，可产生下述结果：

$$\Lambda_{(i)}=e_{ii}^{(0)}/w_{ii}^{(0)}=e_{ii}/w_{ii}\quad i=1,2,\cdots,m$$

$$\Lambda_{(i)\cdot 1}=e_{ii}^{(1)}/w_{ii}^{(1)}\quad i=2,\cdots,m$$

$$\Lambda_{(i)\cdot 12}=e_{ii}^{(2)}/w_{ii}^{(2)}\quad i=3,\cdots,m$$

$$\vdots\qquad\qquad\vdots$$

从上述讨论的消去变换的三个性质可以看出，如在逐步判别中采用消去变换，将使前面讨论的有关逐步判别的算法变得既简单又方便，这是因为消去变换对加人变量和剔除变量是同样适用的，并且因为 $T_{ij}(T_{ij}(\boldsymbol{X}))=\boldsymbol{X}$，如果引进一个变量作了消去变换，随后又将它剔除时还可作同样的消去变换，即对 $\boldsymbol{X}$ 连续施行两次相同的消去变换，矩阵又回到该变量未引进前的情况。再由性质 $T_{ii}(T_{jj}(\boldsymbol{X}))=T_{jj}(T_{ii}(\boldsymbol{X}))(i\neq j)$，故如果选进了 r 个变量，则 $\boldsymbol{E}^{(r)}$ 与这些变量选进的先后次序是无关的。另外，因为如果经 f 次消去变换后，最后选中了 r 个变量，不失一般

性仍可假定是前 r 个变量，因为通过消去变换还可以计算出逆矩阵，即 $E_r^{-1}=E^{(r)}$，所以可以用它来计算 Fisher 或 Bayes 判别函数来。

另外，这样以 Wilks 统计量 Λ 的值作为目标函数，用上述逐步选变量的方法求得 Wilks 统计量 Λ 的值的极小不是全局最小，只是一种局部最优解．对变量选择的全局最优解的求法也只能用我们在第五章讨论的模拟退火法、遗传算法或基于数论方法的序贯优化法来求得．当然，代价就是计算更复杂，计算时间更长。

以上讨论的几种方法都是经典的参数方法，下面介绍几种非参数方法，如线性学习机、K 邻域判别法、势函数判别法、人工神经网络判别法。

五、线性学习机

【基本思路】

线性学习机（Linear Learning Machine）又称线性判别函数法（Linear discrimination function method）。设在训练集中有两类样本，记为 ω^1 和 ω^2．线性判别函数法的目标是找到一个矢量 $\boldsymbol{w}$，如果它们是线性可分的，则总可找到一个矢量 $\boldsymbol{w}$，使得

$$\boldsymbol{x}_k \in \omega^1,\quad \boldsymbol{w}^t\boldsymbol{x}_k > 0$$

$$\boldsymbol{x}_k \in \omega^2,\quad \boldsymbol{w}^t\boldsymbol{x}_k < 0$$

试考察一个二维模式空间线性可分的例子（图 7-3），ω^1 和 ω^2 可借判决面（discrimination plane）分开．在本例中，判决面是一条直线，这条直线是通过原点的，这一点总是可通过坐标旋转与平移达到。判决面可借与之正交并通过原点的权矢量 $\boldsymbol{w}$ 表述。试计算相应的点积 s，

$$s=\boldsymbol{w}^t\boldsymbol{x}_k= |\boldsymbol{w}|\,\|\boldsymbol{x}_k\|\cos(\alpha) \qquad (7\text{-}45)$$

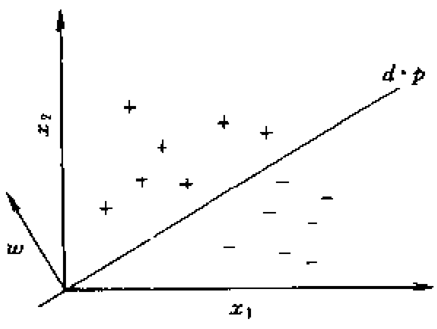


图 7-3　线性分类器示意图

式中，α 是二矢量间的夹角。显然，如 s 是正值，$\boldsymbol{x}_k$ 属 ω^1 类，如 s 是负值，$\boldsymbol{x}_k$ 属 ω^2 类，所以判决面是一个“分类器”。

从这个简单的例子出发，可探求如何求出判决面，即找到一个线性分类器。求判决面的过程，是借助已知的 ω^1 类和 ω^2 类样本进行的，这一过程就称为训练过程，这些已知样本称为训练集。设一组已知分类的样本集，可将其随机分为两部分，一部分作训练集（或称学习集、参比集）用于计算判决面；另一部分作预测集（或称检验集）用于考察找出的判决面能否用来正确地预测训练集以外的样本。

【基本算法】

线性学习机是一种有监督的学习类型的简单线性判别函数的迭代算法，可通过以下步骤来实现。

（1）随机选取一个与样本矢量具有相同维数的矢量作为 $\boldsymbol{w}$。

（2）对于每个样本都进行计算（$k=1,\cdots,n$）：

如果 $\boldsymbol{x}_k\in\omega^1$，而且如果 $\boldsymbol{w}^t\boldsymbol{x}_k>0$，则 $\boldsymbol{w}_{\text{new}}=\boldsymbol{w}_{\text{old}}$，（判决矢量保持不变）；

反之，如果 $\boldsymbol{w}^t\boldsymbol{x}_k<0$，则 $\boldsymbol{w}_{\text{new}}=\boldsymbol{w}_{\text{old}}-\lambda\boldsymbol{x}_k$，（修正判决矢量）；

如果 $\boldsymbol{x}_k\in\omega^2$，而且如果 $\boldsymbol{w}^t\boldsymbol{x}_k<0$，则 $\boldsymbol{w}_{\text{new}}=\boldsymbol{w}_{\text{old}}$，（判决矢量保持不变）；

反之，如果 $\boldsymbol{w}^t\boldsymbol{x}_k>0$，则 $\boldsymbol{w}_{\text{new}}=\boldsymbol{w}_{\text{old}}-\lambda\boldsymbol{x}_k$，（修正判决矢量）；

在这里 $\lambda=2(\boldsymbol{w}_{\text{old}}^t\boldsymbol{x}_k)/\|\boldsymbol{x}_k\|^2$

（3）重复第二步，直至对所有的样本都正确分类。

以上算法是对于线性可分的情况而设计的，对于线性不可分的情况，则规定重复次数，到了规定次数还不能完全将训练集分开，则认为属于线性不可分的。

另外，在本算法中的修正判决矢量的计算，实际上是将当前的不能正确分类的判决矢量进行反射．这是因为

$$
\begin{aligned}
\boldsymbol{w}_{\text{new}}\boldsymbol{x}_k &= (\boldsymbol{w}_{\text{old}} - \lambda \boldsymbol{x}_k)^{t}\boldsymbol{x}_k \\
&= \boldsymbol{w}_{\text{old}}^{t}\boldsymbol{x}_k - 2(\boldsymbol{w}_{\text{old}}^{t}\boldsymbol{x}_k)(\boldsymbol{x}_k^{t}\boldsymbol{x}_k)/\|\boldsymbol{x}_k\|^2 \\
&= -\boldsymbol{w}_{\text{old}}^{t}\boldsymbol{x}_k
\end{aligned}
$$

也就是说，经过这样的修正以后，原来不能正确分类的现在可以正确分类了。一般说来这样的重复次数在 20 次左右就足够了。

六、*K*-最近邻法

【基本思想】

K-最近邻法（*K*-nearest neighbor classification method，*K*-NN 法）是一种直接以模式识别的基本假设——同类样本在模式空间相互较靠近——为依据的分类方法。它计算在最近邻域中 k 个已知样本到未知的待判别样本的距离，即使所研究的体系线性不可分，此法仍可适用。K 最近邻法从算法上讲极为直观，在这种方法中，实际上是要将训练集的全体样本数据存储在计算机内，对每一个待判别的未知样本，逐一计算与各训练集样本之间的距离，找出其中最近的 K 个进行判别。

如 $K=1$，很自然这一个最近邻样本属于何类，未知样本即判属该类。如 $K>1$，则这 K 个最近邻样本不一定都属于一类。采用“表决”的办法，对这 K 个最近邻样本的归属情况，按少数服从多数进行判决。一个近邻相当于一票，但应考虑对各票进行加权，因距离最近的近邻的类属，应予以较重的权。例如按（7-46）式计算：

$$V_{\text{sample}} = \sum V_i / D_i \tag{7-46}$$

在这里，如果 $\boldsymbol{x}_i \in \omega^1$，则取 $V_i=1$；反之，如果 $\boldsymbol{x}_i \in \omega^2$，$V_i=-1$；$D_i$ 是待判别的样本与近邻的距离。这样，如果求得的 V_{sample} 为正，则可认为 $\boldsymbol{x}_i$ 属于 ω^1，反之，$\boldsymbol{x}_i$ 属于 ω^2，即，

$$\text{当 } V_{\text{sample}}>0\text{，则 } \boldsymbol{x}_{\text{sample}} \in \omega^1\text{；}$$

$$\text{而 } V_{\text{sample}}<0\text{，则 } \boldsymbol{x}_{\text{sample}} \in \omega^2$$

D_i 的作用相当于一个权因子，即如果近邻 j 与样本 $\boldsymbol{x}_i$ 的距离很小时，它权值就大，而那些距离大的近邻就权值较小。

【基本算法】

K-最近邻法（*K*-NN 法）算法相当简单，可由以下步骤来完成：

（1）取一个未知样本，记为 $\boldsymbol{x}_{\text{unknown}}$，计算该样本到训练集各样本的距离 D_i（$i=1,2,\cdots,n$），在此 n 为所有训练集样本的总数。

（2）取出 k 个距离最短的训练集样本，计算它们的权值和：

$$V_{\text{unknown}} = \sum V_i / D_i \qquad (i=1,2,\cdots,k)$$

如果 $\boldsymbol{x}_i \in \omega^1$，则取 $V_i=1$；反之，如果 $\boldsymbol{x}_i \in \omega^2$，$V_i=-1$；$D_i$ 是待判别的样本与近邻的距离。

（3）建立判别标准，即当

$$V_{\text{unknown}}>0\text{，则 } \boldsymbol{x}_{\text{unknown}} \text{ 判为第一类 } \omega^1\text{；}$$

反之，$V_{unknown}<0$，则 $\boldsymbol{x}_{unknown}$判为第二类 ω^2

关于此法中 k 值取多少，文献中多有讨论[5]，但很难有一定论，一般还是靠经验来定。此外，有人还提出了一种交替 KNN（Alternative KNN）方法，读者可参阅文献［6～8］。

七、势函数判别法

【基本思想】

势函数分类法（classification with potential function）实际上是 K-最近邻法（K-NN 法）的一种很有效的拓广变种。它把每一个已知种类样本的代表点都看成某种“势场”的源，例如一个电场源即一个点电荷。在距某点距离为 D 处的场强 $V(D)$ 可用下列一类公式表示：

$$V(D)=1/D \tag{7-47}$$

$V(D)$也可采用更复杂的形式，如

$$V(D)=1/(1+qD^2) \tag{7-48a}$$

或

$$V(D)=e^{-D^2 q} \tag{7-48b}$$

式中，q 为可调参数。

所有点的场均分布于整个空间且互相重叠。对于未知点属于哪一类的判断基于这样一个原则：未知点属于在该点有最大势场强度的哪一类，即用下列判据，

$$V_{sample}=(1/n_1)\sum V(\boldsymbol{x}_{j1})-(1/n_2)\sum V(\boldsymbol{x}_{j2}) \tag{7-49}$$

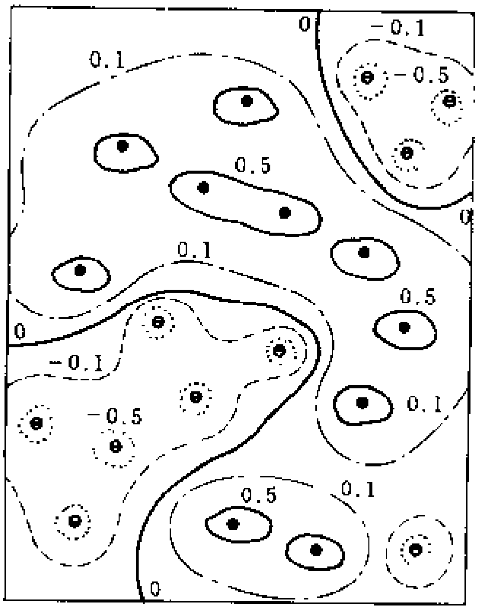


图 7-4　势函数法示意图

式中，$V(\boldsymbol{x}_{j1})$和 $V(\boldsymbol{x}_{j2})$分别代表第一类点和第二类点对未知样本点处施加的势场，n_1、n_2 分别为第一、第二类点数目。V_{sample}为未知点处两类点平均作用势的差值。当 $V_{sample}>0$，则 $\boldsymbol{x}_{sample}\in\omega^1$；而当 $V_{sample}<0$，则 $\boldsymbol{x}_{sample}\in\omega^2$。$V_{sample}=0$ 时为分界面，一般为复杂的不连续的曲面。图 7-4 表示这类分界面的示意图。

【基本算法】

势函数分类法算法也很简单，可由以下步骤来完成：

(1) 取一个未知样本，记为 $\boldsymbol{x}_{unknown}$，计算该样本在已知训练集场中的场强 $V_{unknown}(D)$，即

$$V_{unknown}(D)=\sum 1/(1+q_i D_i^2)(i=1,2,\cdots,n) \tag{7-50}$$

在此 n 为所有训练集样本的总数，D_i^2 为该样本到训练集各样本的距离，q_i 为预置常数，当训练集样本 i 属于第一类时取正值，而当训练集样本 i 属于第二类时取负值。

(2) 建立判别标准，即

当 $V_{unknown}(D)>0$，则 $\boldsymbol{x}_{unknown}$，判为第一类 ω^1；

反之，$V_{unknown}(D)<0$，则 $\boldsymbol{x}_{unknown}$判为第二类 ω^2

势函数法的一个较简单的变种是只考虑一批近邻的势的方法，例如，可只考虑K个近邻，取判别函数为

$$V_{\text{unknown}}(D)=\sum 1/(1+q_iD_i^2)(i=1,2,\cdots,k) \tag{7-51}$$

这就与 K-近邻法相似了。

八、人工神经网络判别法

【基本思想】

人工神经网络方法在前一章的非线性校正方法已作出了一般性的介绍，在此只对其在化学模式识别中的应用给出必要的补充介绍。人工神经网络由于其具有非线性映照的能力，用于化学模式识别应该说是有一定优势的，近年来在化学模式识别中得到了相当广泛的应用[9~11]，本节仍以多层前传网络误差反传算法为主来作出介绍。

人工神经网络用于化学模式识别的基本思路与前述的判别分析方法是相同的，它也是通过对已知类别的训练集的拟合，来建立模型来进行分类与预测的。它与第六章中讨论的非线性校正方法的区别仅在于校正方法的拟合为连续变化函数，而化学模式识别则是以 0，1 或 −1、+1 为其输出来进行拟合建模的。

【基本算法】

用于化学模式识别的误差反传算法算法将由以下几步构成：

(1) 随机选取网络之间的连接权重 $w_{kj}^{m_2}$和 $w_{ji}^{m_1}$（$j=1$，…，n；$i=1$，…，m；$k=1$，…，L)，其中 n 为输入矢量的节点数（或称输入矢量的维数）；m 为隐节点数；L 为输出节点数。一般就用（−0.5～0.5）的均匀分布的随机函数产生。

(2) 设$a=1$，…，A，重复迭代进行以下步骤，直至收敛：

① 前传计算：按照下面两式来计算 $\boldsymbol{O}$，

$$\begin{aligned}O_k&=f(\sum w_{kj}^{m_2}y_j^m+w_0^{m_2})\\&=f(\sum w_{kj}^{m_2}f(\sum w_{ji}^{m_1}I_i+w_0^{m_1}I_i)+w_0^{m_2})\end{aligned} \tag{7-52}$$

式中，$y_j^m=f(\sum w_{ji}^{m_1}I_i+w_0^{m_1})$，活性函数或取符号函数，即 $f(u)=\text{sgn}(u)=f_h(u)$，或取非线性反正弦函数 $f(u)=\tanh(u)$。与所期望得到的数值比较，按式（7-53）计算出误差函数 E。

$$E=1/2(\sum(O_k-E_k)^2) \tag{7-53}$$

② 反传调节：计算出由式（7-53）表出的目标函数对连接权重的一阶导数，并以此来调节连接权重 $w_{kj}^{m_2}$和 $w_{ji}^{m_1}$。(因在此只介绍最陡梯度法，所以用的是一阶导数，别的优化算法同样是可行的)。

$$\Delta w_{ij}=\rho\delta_i u_j \tag{7-54}$$

式中，ρ 为一步长因子，可在（0～1）之间取值；u 为第 j 节点上的抽象变量，一般视其是在输出层还是隐蔽层，就分别等于 $f(\sum w_{ji}^{m_1}I_i+w_0^{m_1})$或者 $f(\sum w_{kj}^{m_2}y_j^m+w_0^{m_2})$；$\delta$ 为梯度因子，对于不同层，有不同表达式：

$$\delta_i=\begin{cases}(u_i-E_i)(u_i)(1-u_i) & u_i\text{ 为输出节点} \quad (7\text{-}55a)\\(\sum w_{ij}^m\delta_k)(u_i)(1-u_i) & u_i\text{ 为隐蔽层节点} \quad (7\text{-}55b)\end{cases}$$

在这里 $u(1-u)=u'$。因为

$$u=f(\sum w_{ij}^{m_1}I_i+w_0^{m_1})=f(x)=1/(1+e^{-x}) \tag{7-56}$$

所以有

$$\begin{aligned}u'=f''(x)&=(1/(1+e^{-x}))/dx=-(1+e^{-x})^{-2}(e^{-x})(-1)\\&=(1/(1+e^{-x}))(1-(1/(1+e^{-x})))=u(1-u)\end{aligned}$$

式（7-54）中步长因子 ρ 的取值大小对收敛速度有很大的影响，如果取值太大，则可能引起迭代过程的振荡；反之，若取值太小则会导致权重调节的迭代过程收敛太慢。一般说来，为了加快迭代过程且防止迭代过程的振荡，一般是在式中引入一个惯量因子 λ，

$$\Delta w_{ij}(\alpha+1)=\rho\delta_j u_j+\lambda\Delta w_{ij}(\alpha) \tag{7-57}$$

以保证迭代收敛速度。

尽管人工神经网络的研究已取得令人兴奋的进展，但是，总的说来对它的研究无论在理论还是应用上都还处于初期阶段，还存在很多问题值得探讨，如网络训练速度、过拟合、全局寻优、网络结构与网络能力等问题都存在未知因素，值得进一步研究。读者可参阅文献［12～15］。

第五节　无监督的模式识别方法——聚类分析法

聚类分析是多元统计中的一种实用性很强的方法，其主要思路就是利用同类样本应彼此相似，即常说的“物以类聚”，相类似的样本在多维空间中彼此的距离应小些，而不同类的样本在多维空间中彼此的距离应大些。聚类分析即为如何使相似的样本“聚”在一起，从而达到分类的目的。一般说来，聚类分析方法可分为凝聚法与分离法两类[16]，凝聚法开始时将每个样本认作一个类，然后由这些类逐步凝聚成较大的类，下面将讨论的系统聚类法就是凝聚法的例子；而分割法则与之相反，开始时将所有的样本认作一个大类，在逐步分割为较小的类，以后将介绍的一分为二法是分割法的一个例子。

一、系统聚类法

【基本思想】

系统聚类法（Hierarchical Clustering Method）亦称谱系聚类法，是一种很常用的凝聚法方法，其基本思想是：先将 n 个样本各自看成一类，然后规定样本之间的距离和类与类之间的距离。开始，因每个样本自成一类，类与类之间的距离与样本之间的距离是等价的，选择距离最小的一对并成一个新类，计算新类与其他类的距离，再将距离最小的两类并成一类，这样每次减少一类，直至所有的样本都成为一类为止。

类与类之间的距离有许多定义的方法[17]，它与前述样本与样本之间的距离是有区别的，例如定义类与类之间的距离为两类之间样本中的最近的距离，或者定义类与类之间的距离为两类重心之间的距离等等，不同的定义就产生了系统聚类的不同方法，本节将对常用的 8 种定义类与类之间距离的不同方法进行简要介绍。为区别样本之间的距离与类与类之间的距离，在以后的讨论中，凡是用小写字母 d 表示的为样本之间的距离，而用大写字母 D 表示的为类与类之间的距离。

（一）最短距离法

最短距离法定义类 G_i 与 G_j 之间的距离为

$$D_{ij}=\min_{x_k\in G_i,\,x_l\in G_j}\{d_{kl}\} \tag{7-58}$$

式中，d_{kl}是样本 $\boldsymbol{x}_k$ 与 $\boldsymbol{x}_l$ 的距离，即类与类之间的距离为两类之间最近的样本之间的距离。

【基本算法】

定义了上述的类与类之间的距离，基于最短距离的系统聚类法的算法可由下述计算步骤组成：

(1) 规定样本之间的距离(可为前述的欧氏距离或其他距离),计算样本两两距离的对称阵,记为 $D_{(0)}$,开始每个样本自成一类,这时显然 $D_{pq}=d_{pq}$。

(2) 选择 $D_{(0)}$的最小元素,设为 D_{pq},则将 G_p 和 G_q 合并成一新类,记为 G_r,则,$G_r=\{G_p,G_q\}$。将 $D_{(0)}$中 p、q 行及 p、q 列并成一个新行新列,新行新列对应 G_r,所得的矩阵记作 $D_{(1)}$。

(3) 计算新类与其他类的距离

$$
\begin{aligned}
D_{rk} &= \min_{x_i \in G_r, x_j \in G_k}\{d_{ij}\} = \min\{\min_{x_i \in G_p, x_j \in G_k}\{d_{ij}\}, \min_{x_i \in G_q, x_j \in G_k}\{d_{ij}\}\} \\
&= \min(D_{pk}, D_{qk})
\end{aligned}
$$

(4) 对$D_{(1)}$重复上述对 $D_{(0)}$的计算得 $D_{(2)}$,如此下去直到所有的元素成为一类为止。

(二) 最长距离法

最长距离法定义类 G_i 与 G_j 之间的距离为

$$
D_{ij} = \max_{x_k \in G_i, x_l \in G_j}\{d_{kl}\} \tag{7-59}
$$

式中,d_{kl}是样本 $\boldsymbol{x}_k$ 与 $\boldsymbol{x}_l$ 的距离,即类与类之间的距离为两类之间最远的样本之间的距离。

由于最长距离法的基本算法与最短距离法是完全类似的,所以,在此只给出计算新类与其他类的递推公式:

$$
\begin{aligned}
D_{rk} &= \max_{x_i \in G_r, x_j \in G_k}\{d_{ij}\} \\
&= \max\{\max_{x_i \in G_p, x_j \in G_k}\{d_{ij}\}, \max_{x_i \in G_q, x_j \in G_k}\{d_{ij}\}\} \\
&= \max(D_{pk}, D_{qk})
\end{aligned} \tag{7-60}
$$

(三) 中间距离法

从上述对最短距离法和最长距离法的讨论可知,它们的差别只在于两点:① 类与类之间距离的定义不同;② 计算新类与其他类的距离所用的递推公式不同。下面将介绍其他系统聚类法之间的差异也都在这两方面,故下述讨论将主要介绍新类与其他类距离的递推公式。

中间距离法定义的类与类之间的距离既不采用两类中样本之间的最短距离,也不采用两类中样本之间的最长距离,而是采用最近和最远之间的距离。

如果在某一步将类 G_p 与 G_q 合并为 G_r,任一类 G_i 与 G_r 之间距离的取法可由如下三角形说明(参见图 6-6)。如果用最短距离法,D_{ir}等于 D_{ip}和 D_{iq}中短的一边;如果用最长距离法,D_{ir}等于 D_{ip}和 D_{iq}中长的一边。而在中间距离法中,D_{ir}既不取 D_{ip}也不取 D_{iq},D_{ir}取的是夹在中间的三角形中线,由初等几何知

$$
\text{三角形中线} = [(1/2)D_{ip}^2 + (1/2)D_{iq}^2 - (1/4)D_{pq}^2]
$$

故可取

$$
D_{ir}^2 = [(1/2)D_{ip}^2 + (1/2)D_{iq}^2 - \beta D_{pq}^2] \qquad -1/4 \leqslant \beta \leqslant 0 \tag{7-61}
$$

当 $\beta=-1/4$ 时,D_{ir}就是上述的三角形中线。(7-61) 式也就是中间距离法的递推公式。

(四) 重心法

从物理的观点看,一个类用它的重心(该类样本的均值向量)做代表比较合理,类与类之间的距离就用重心之间的距离来代表。设 G_p 与 G_q 的重心分别为 $\bar{x}_p$ 和 $\bar{x}_q$,G_p 与 G_q 的距离为

$$
D_{pq} = d_{\bar{x}_p \bar{x}_q} \tag{7-62}
$$

对应于这种定义的系统聚类法叫重心法。

设某一步将类 G_p 与 G_q 合并为 G_r,它们各含有 n_p,n_q,$n_r(n_r=n_p+n_q)$个样本,则此种方

法的递推公式为

$$D_{ir}^2=[(n_p/n_r)D_{ip}^2+(n_q/n_r)D_{iq}^2-(n_pn_q)/(n_r^2)D_{pq}^2] \tag{7-63}$$

（五）类平均法

重心虽有很好的代表性，但未充分利用各样本的信息，有人建议将两类之间的距离定义为这两类元素所有样本两两之间的平均平方距离，即

$$D_{pq}^2=(1/n_pn_q)\sum_{x_i\in G_p,x_j\in G_q}\sum d_{x_ix_j}^2 \tag{7-64}$$

在此，n_p 和 n_q 分别为类 G_p 与 G_q 中的样本数。容易导出[3]，此种方法的递推公式为

$$D_{ir}^2=[(n_p/n_r)D_{ip}^2+(n_q/n_r)D_{iq}^2] \tag{7-65}$$

有人认为类平均法是系统聚类法中比较好的方法之一。

（六）可变类平均法

因在类平均法的递推公式中没有反映 D_{pq}^2 的影响，故有人建议将（7-65）递推公式改为

$$D_{ir}^2=[(n_p/n_r)(1-\beta)D_{ip}^2+(n_q/n_r)(1-\beta)D_{iq}^2+\beta D_{pq}^2] \tag{7-66}$$

式中，$\beta<1$，对应于这个递推公式的系统聚类法叫做可变类平均法。

（七）可变法

如将上述讨论的递推公式改为

$$D_{ir}^2=[(1-\beta)/2](D_{ip}^2+D_{iq}^2)+\beta D_{pq}^2 \tag{7-67}$$

式中，$\beta<1$，那么，对应于这个递推公式的系统聚类法叫做可变法。

可变类平均法和可变法的分类效果与 β 的选择关系极大，有一定的人为任意性，因此在实际使用不太多。

（八）方差平方和法

这个方法是由 Ward 提出，故也有 Ward 法之称。该方法的主要思想是来源于方差分析，即如果类分得正确，同类样本的方差平方和应当较小，类与类之间的方差平方和应当较大。

设有 n 个样本拟分成 k 类 G_1，G_2，…，G_k，$\boldsymbol{x}_{it}$表示 G_t 中的第 i 个样本，每个样本都是一个 m 维向量，n_t 表示 G_t 中的样本个数，$\overline{\boldsymbol{x}}^{(t)}$为 $G_t(t=1,2,\cdots,k)$的重心，亦即该类的均值向量，则 G_t 的样本方差平方和为

$$S_t=\sum_{i=1}^{n_t}(\boldsymbol{x}_{it}-\overline{\boldsymbol{x}}^{(t)})'(\boldsymbol{x}_{it}-\overline{\boldsymbol{x}}^{(t)})$$

$$S=\sum_{t=1}^{k}\sum_{i=1}^{n_t}(\boldsymbol{x}_{it}-\overline{\boldsymbol{x}}^{(t)})'(\boldsymbol{x}_{it}-\overline{\boldsymbol{x}}^{(t)})=\sum_{t=1}^{k}S_t$$

当 k 固定时，要选择使 S 达到极小的分类。n 个样本，分成 k 类，一切可能的分法可由下式给出

$$R(n,k)=(1/k!)\sum_{i=0}^{k}(-1)^{k-i}\binom{k}{i}i^n \tag{7-68}$$

有关此式的证明可参阅文献［3］。上式说明，此类优化问题为组合优化问题，求全局最优解是很困难的一个问题，Ward 法给出的是一个局部最优解。其基本思路为：先将 n 个样本各自分成一类，此时 S 最小，然后每次缩小一类，因每缩小一类后总方差平方和 S 就要增大，选择使 S 增加最小的两类进行合并，直至所有的样本归为一类为止。

从上述讨论中看，似乎方差平方和法与前面介绍的 7 种系统聚类法有较大的差异，其实可以证明，方差平方和法的类距离计算的递推公式为

$$D_{ir}^2=[(n_i+n_p)/(n_i+n_r)]D_{ip}^2+[(n_i+n_q)/(n_i+n_r)]D_{iq}^2-[(n_i)/(n_i+n_r)]D_{pq}^2 \quad (7\text{-}69)$$

上述所介绍的 8 种系统聚类法，并类的原则和步骤是完全一样的。所不同的是类与类之间的距离有不同的定义，从而得到不同的递推计算公式。Wishart[18]发现它们的递推公式可以统一起来，它们统一的形式为

$$D_{ir}^2=\alpha_p D_{ip}^2+\alpha_q D_{iq}^2+\beta D_{pq}^2+r\,|D_{ip}^2-D_{iq}^2| \quad (7\text{-}70)$$

方法不同，式中的 α_p、α_q、β 和 r 的取值也不同。表 7-2 列出了上述 8 种系统聚类法 4 个参数的取值。递推公式的统一使八种方法的共性完全统一起来了，为编制计算机统一程序提供了极大的方便，文献 [19] 给出了这样的程序。

表 7-2　系统聚类法参数表

方　　法	α_p	α_q	β	r
最短距离法	1/2	1/2	0	$-(1/2)$
最长距离法	1/2	1/2	0	1/2
中间距离法	1/2	1/2	$-(1/2)$	0
重心法	n_p/n_r	n_q/n_r	$(n_pn_q)/(n_r^2)$	0
类平均法	n_p/n_r	n_q/n_r	0	0
可变类平均法	$(1-\beta)/2$	$(1-\beta)/2$	β	0
可变法	$(1-\beta)n_p/n_r$	$(1-\beta)n_q/n_r$	β	0
方差平方和法	$(n_i+n_p)/(n_i+n_r)$	$(n_i+n_p)/(n_i-n_r)$	$-(n_i)/(n_i+n_r)$	0

【示例】

现以一个最简单的例子来说明上述系统聚类法的聚类过程。表 7-3 给出了一个薄层色谱的 5 个溶剂系统（分别记为 A、B、C、D、E）分离 4 种物质（记为 a、b、c、d）的保留时间比移值 $hR_f(R_f\times100)$数据。

表 7-3　5 个 TLC 体系的比移值 hR_f

溶剂 / 分离物	A	B	C	D	E
a	100	80	80	40	50
b	80	60	70	20	10
c	70	50	40	20	20
d	60	40	50	10	10

今欲将 TLC 体系分类，如欲找出 A～E 诸溶剂中哪些是相似的，哪些是不相似的，这种信息对选择适宜的溶剂体系进行二维 TLC 分离有意义。首先将 A、B、C、D、E 各认作独立的类，由表 7-3 可计算距离阵，在此采用的是欧氏距离，结果示于表 7-4。

表 7-4　TLC 溶剂体系的欧氏距离

溶剂系统	A	B	C	D	E
A	0				
B	40	0			
C	38.7	17.3	0		
D	110.4	70.7	78.1	0	
E	111.4	72.1	80.6	14.1	0

如采用最短距离法，这 5 类之间的最小距离为 DE，即 D 与 E 是最相似的溶剂体系。可先将 DE 合并为一类，记为 D^*，表 7-5 列出了 DE 聚为一类 D^* 后的欧氏距离，从表 7-5 可以看出，当前类与类间最小的距离为 BC，将它们再聚为一类，并记为 B^* 后的欧氏距离表示于

表 7-6，最后将 A 与 B^* 合并为一类 A^*，其类间距离表示于表 7-7。

表 7-5　最短距离法 DE 聚为一类 D^* 后的欧氏距离

溶剂系统	A	B	C	D^*
A	0			
B	40	0		
C	38.7	17.3	0	
D^*	110.4	70.7	78.1	0

表 7-6　最短距离法 DE、BC 聚类后的欧氏距离

溶剂系统	A	B^*	D^*
A	0		
B^*	38.7	0	
D^*	110.4	70.7	0

表 7-7　最短距离法 A 与 B^* 聚类后的欧氏距离

溶剂系统类别	A^*	D^*
A^*	0	
D^*	70.7	0

上述结果可绘成谱系图（图 7-5），由谱系图可得到 TLC 体系的自然分类，这对研究 TLC 的作用机制有一定意义。如欲将溶剂体系分为两类，以便设计一个二维 TLC 系统，从两类溶剂中可取一种分别用于垂直的两个方向展开，可将最高的连接线切断，即 A、B、C 是一类，D、E 是另一类。

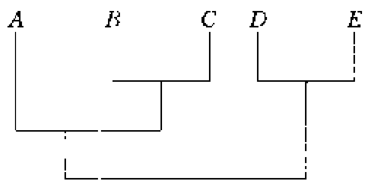


图 7-5　TLC 溶剂体系的最短距离法聚类谱系图

同理，可对此 TLC 溶剂体系用其他的系统聚类的方法来进行分析，下面列出最长距离法、重心法、方差平方和法的结果和聚类谱系图，以便读者理解这些方法的运算过程。

对于最长距离法，首先因为这 5 类之间的最小距离为 DE，即 D 与 E 是最相似的溶剂体系，也是将 DE 合并为一类，记为 D^*，表 7-8 列出了 DE 聚为一类 D^* 后的欧氏距离。从表 7-8 可以看出，当前类与类间最小的距离为 BC，将它们再聚为一类，并记为 B^* 后的欧氏距离表示于表 7-9，最后将 A 与 B^* 合并为一类 A^*，其类间距离表示于表 7-10。

表 7-8　最长距离法 DE 聚为一类 D^* 后的欧氏距离

溶剂系统	A	B	C	D^*
A	0			
B	40	0		
C	38.7	17.3	0	
D^*	111.4	72.1	80.6	0

表 7-9　最长距离法 DE、BC 聚类后的欧氏距离

溶剂系统	A	B^*	D^*
A	0		
B^*	40	0	
D^*	111.4	72.1	0

表 7-10　最长距离法 A 与 B^* 聚类后的欧氏距离

溶剂系统	A^*	D^*
A^*	0	
D^*	111.4	0

上述结果可绘成谱系图（图 7-6），此结果与最短距离法是完全一致的，只是在类间距离有所不同。

对于重心法，首先也是将 DE 并为一类，记为 D^*，表 7-11 列出了 DE 聚为一类 D^* 后的欧氏距离。

注意到在表 7-11 中，C 点与新类 D^* 间的距离，是原 CD 与 CE 间距离的均值。聚类过程继续进行，从表 7-11 可以看出，当前类与类间最小的距离为 BC，将它们再聚

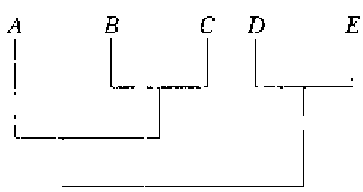


图 7-6　TLC 溶剂体系的最长距离法聚类谱系图

为一类，并记为 B^* 后的欧氏距离表示于表 7-12，最后将 A 与 B^* 合并为一类 A^*，其类间距离表示于表 7-13。

表 7-11　重心法 DE 聚为一类 D^* 后的欧氏距离

溶剂系统	A	B	C	D^*
A	0			
B	40	0		
C	38.7	17.3	0	
D^*	110.9	71.4	79.3	0

表 7-12　重心法 DE、BC 聚类后的欧氏距离

溶剂系统	A	B^*	D^*
A	0		
B^*	39.3	0	
D^*	110.9	75.3	0

表 7-13　重心法 A 与 B^* 聚类后的欧氏距离

溶剂系统	A^*	D^*
A^*	0	
D^*	93.1	0

上述结果可绘成谱系图（图 7-7），此结果与前述方法也是完全一致的。

方差平方和法的步骤与上述方法略有不同，其过程如下：首先将 A、B、C、D、E 仍看成 5 类，此时总方差 $S=0$，然后将其中任意两类合并，并计算它们合并后的方差平方和，例如将 A、B 合并，所得的方差平方和为

$$S_{AB}=\sum_{i=1}^{2}(\boldsymbol{x}_{AB,i}-\overline{\boldsymbol{x}}^{(AB)})'(\boldsymbol{x}_{AB,i}-\overline{\boldsymbol{x}}^{(AB)}) \qquad (7\text{-}71)$$

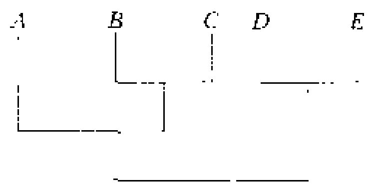


图 7-7　TLC 溶剂体系的重心法聚类谱系图

式中

$$\overline{\boldsymbol{x}}^{(AB)}=\begin{bmatrix}(100+80)/2\\(80+60)/2\\(70+50)/2\\(60+40)/2\end{bmatrix}=\begin{bmatrix}90\\70\\60\\50\end{bmatrix}$$

$$\boldsymbol{x}_{AB,1}=\begin{bmatrix}100\\80\\70\\60\end{bmatrix} \qquad \boldsymbol{x}_{AB,2}=\begin{bmatrix}80\\60\\50\\40\end{bmatrix}$$

$$S_{AB}=400+400=800$$

如将 B、C 合并，所得 $S_{BC}=150$，一切可能两两并类增加的方差平方和列于表 7-14，以 ED 合并 S 增加最少，故可将它们合并，然后，再计算 ED 与其他未并类的 A、B、C 进行并类，发现所增加的方差较多，而将 BC 并类使方差增加较少，故可先将 BC 并类，然后再考虑所有可能的并类，重复上述步骤，直至所有样本归为一类为止。对 TLC 溶剂体系的方差平方和法并类情况示于表 7-15，它很类似于其他系统聚类的方法，所得结果也相同。从上述讨论可知，在此介绍的 8 种系统聚类方法的确具有一定共性的。

表 7-14　两两并类后 S 的增加量

溶剂系统	S_{AX}	S_{BX}	S_{CX}	S_{DX}	S_{EX}
A	0				
B	800	0			
C	750	150	0		
D	6010	2500	1425	0	
E	6200	2600	3250	100	0

表 7-15 TLC 溶剂体系的方差平方和法并类情况

分类数目	类	S	分类数目	类	S
5	A、B、C、D、E	0	2	ABC、DE	1233.3
4	A、B、C、DE	100	1	ABCDE	10200
3	A、BC、DE	250			

二、一分为二法

系统聚类是将类由多到少，另一种分类的思想是分解法，则是使类由少到多，开始全体样本全为一类，然后分成两类、三类，…直至所有的样本各自成一类。一分为二法就是这样一种分解方法。

【基本思路】

一分为二法是将某一类分解成为两子类的方法，然后对其子类又可一分为二。其基本思路是在不断的分解过程中，使得类间方差保持最大。如设某类 G 中有 n 个样本，它的两个子类 G_1 和 G_2 各有 n_1 和 n_2 个样本，两类的重心，即均值向量分别为 $\bar{\boldsymbol{x}}^{(1)}$ 和 $\bar{\boldsymbol{x}}^{(2)}$，而类 G 的重心为 $\bar{\boldsymbol{x}}$，则类 G、G_1 和 G_2 的方差平方和分别为

$$S = \sum_{k=1}^{2}\sum_{i=1}^{n_k}(\boldsymbol{x}_{ki} - \bar{\boldsymbol{x}})^{t}(\boldsymbol{x}_{ki} - \bar{\boldsymbol{x}})$$

$$S_k = \sum_{i=1}^{n_k}(\boldsymbol{x}_{ki} - \bar{\boldsymbol{x}}^{(k)})^{t}(\boldsymbol{x}_{ki} - \bar{\boldsymbol{x}}^{(k)}) \qquad k = 1,2$$

在此，$\boldsymbol{x}_{ki}(k=1,2,i=1,2,\cdots,n_k)$为样本向量，$S$、$S_1$ 和 S_2 分别为类 G 和它的两个子类 G_1 和 G_2 的方差平方和。如果类分解得合理，应使 S_1+S_2 尽可能的小，或使$(S-S_1-S_2)$尽可能的大。如记 $E=(S-S_1-S_2)$，则 E 是此一分为二的分解过程中的目标函数。根据方差分解定理，可得

$$E=(S-S_1-S_2)=[(n_1n_2)/n](\bar{\boldsymbol{x}}^{(1)}-\bar{\boldsymbol{x}}^{(2)})^{t}(\bar{\boldsymbol{x}}^{(1)}-\bar{\boldsymbol{x}}^{(2)})$$

或者

$$E=(S-S_1-S_2)=[(nn_1)/n_2](\bar{\boldsymbol{x}}^{(1)}-\bar{\boldsymbol{x}})^{t}(\bar{\boldsymbol{x}}^{(1)}-\bar{\boldsymbol{x}})$$

如类 G 中有 n 个样本，则一切可能的分法有 2^n-1 种，是一个组合优化问题，一般说来，求全局最优比较困难，一般是求得一个局部最优解。

【基本算法】

在此，介绍一种局部最优解的算法，它由下述步骤组成：

(1) 首先将 n 个样本都认作为类 G_1 的，依次从中取一个样本，将其划入 G_2，计算此时的目标函数 E。

(2) 比较依次计算所得的 n 个 E，并从中找到最大者，记为 $E(1)$，作为一次成功分解。

(3) 继续分解，即从类 G_1 所剩 (n-1) 样本中再取一个样本划入 G_2，并计算此时的目标函数 E，比较依次计算所得的 ($n-1$) 个 E，并从中找到最大者，记为 $E(2)$，认作一次成功分解。重复此过程，可得一系列最大的 E，记为 $E(3),E(4),\cdots,E(k)$，它们分别表示在 G_2 中含 2 个样本、3 个样本、…、k 个样本的最大 E 值。

(4) 找到一个自然数 k^*，使得

$$E(k^*)=\max E(k) \qquad (k=1,2,\cdots,n)$$

(5) 前 k^* 次进入 G_2 的样本为一类，其余 ($n-k^*$) 个样本为一类，一分为二的分解完毕。

(6) 继续对所得的 G_1 和 G_2 进行一分为二的分解，直至所有的 n 个样本独自成为一类为止。

一分为二法不需大量内存，计算量也不大，因此是一种较好的聚类分法。值得提出的是，上述一分为二法是利用类间方差极大为其目标函数，用其他目标函数也是可行的。文献［22］就介绍了一种基于欧氏距离的一分为二的分解法。

三、最小生成树法

【基本概念与基本思路】

最小生成树（Minimal Spanning Tree Method，MST）是图论中的一种算法。也可用来进行聚类计算。在介绍最小生成树方法之前，似有必要介绍一下有关图论和树的基本概念。

图由结点的集合 V 和边的集合 E 组成：

$$V=\{v_1,v_2,v_3,\cdots,v_n\} \qquad n \text{ 为结点的数目}$$

$$E=\{e_1,e_2,e_3,\cdots,e_m\} \qquad m \text{ 为边的数目}$$

即图 G 可表示为

$$G=\{V,E\}$$

在一个图中边的旁侧可附加数字以描述此边的某种数量特征，这称为边的权。两点之间通过一系列的边联系起来，这些边称作链，一个链如果是封闭的则称这个链组成一个回路，如果 n 个点之间都有链互相联结，则称这些点和链组成了一个联结图，没有回路的联结图叫做树。如果树包含了 n 个点，则称这个树为联结图的支撑树，树的所有边的长度之和叫做树的重量。在联结图中具有最轻重量的支撑树叫做最小生成树，亦称最小支撑树。

最小生成树可按 Kruskal 算法找出，这个算法十分简单，要求将权值最小的边绘人树中，但不得与已有的边形成回路。

【基本算法】

① 先计算各点间的距离；

② 连接最小距离的两点，然后连接次小距离的两点；

③ 继续连接所剩下点的最小距离的两点，并检查是否形成回路，如形成回路则取消连接，否则，继续连接，直至所有的点都被连接；

④ 对连接所得的树进行检查，找到最大距离的边，将其割断，即得两类，如此继续分割，直至类数已达到所要分的类数。

如对前述的 TLC 溶剂体系进行最小生成树分类，可得与系统聚类法完全一致的结果。文献［3］对系统聚类法中的最小距离法与最小生成树的关系进行了较详细的讨论，并证明了最短距离法是在由最小生成树为基点而定义的误差函数下的最合理的聚类方法。

四、*K*-均值聚类法

用系统聚类法聚类，样品一旦划人某个类以后就不变了，这要求分类的方法比较准确，此外，系统聚类法一般要存人距离矩阵，当 n 较大时占用的内存太多。计算方法的迭代思想给我们以启发，能否先给一个粗糙的初始分类，然后用某种原则进行修改，直至分类比较合理为止。采用这种思想产生的聚类分析法叫做动态聚类法。为了得到初始分类，有时设法选择一些凝聚点，让样品按某种原则向凝聚点凝聚。动态聚类法大体可用图 7-8 的框图表示。

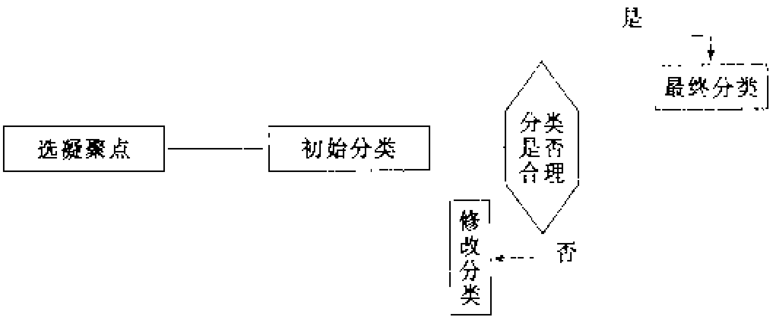


图 7-8 动态聚类法程序框图

图 7 8 的每一部分构成均有很多种方法，这些方法按框图进行组合就可得到很多种动态聚类法。在本节只讨论其中一种，即 K-均值聚类法（K-means Cluster Method），它是一种用得较多的方法。该法最早是由 MacQueen[20]提出，随后许多人对此作了改进。

【基本思路】

K-均值聚类法的基本思路是先人为地定出一个分类数目 k，并任取 k 个点作为初始凝聚点，然后，逐个将别的样本引人，引进一个样本就计算它与 k 个类的距离（与重心的距离），选定距离最小者将其并入该类，再重新计算各类的重心，并以该重心作为新的凝聚点，直至每个样本都有了各自的归类。

【基本算法】

（1）最简单的 K-均值聚类法的计算过程

① 人为地定出一个分类数目 k，取前 k 个点作为初始凝聚点。

② 将剩余的（$n-k$）个样本逐个引入，并计算它与已有 k 个类的距离（与重心的距离），选定距离最小者，则将其并入该类，再重新计算各类的重心，并以该重心作为新的凝聚点。

③ 将 n 个样本再从头至尾逐个输人，再计算它与已有 k 个类的重心的距离，并选定距离最小者，则将其并人该类，再重新计算各类的重心，并以该重心作为新的凝聚点。如果 n 个样本通过后所分的类与原来的类没有改变，则过程停止，否则重复第③步。

从上述计算过程可以看出，此算法计算简单，分类迅速，占用计算机内存小，但由于人为地定了 k 个类，有时定得不合适而影响分类效果，改进的办法是在修改分类的过程中类的数目也可以根据情况有所变化，太近的类可以合并，太远的类可以分离出来产生新类。下面将给出改进后的 K-均值聚类算法。

（2）改进的 K-均值聚类法的计算过程

① 人为地定出三个数目 k、c 和 r。

② 取前 k 个点作为初始凝聚点，计算 k 个凝聚点两两之间的距离，如最小距离$<c$，则将相应的两凝聚点合并，用这两点的重心作为新凝聚点，重复此步骤，直至所有凝聚点之间的距离均大于等于 c 为止。换言之，即要使凝聚点两两之间的距离均大于等于 c，如小于 c，则用系统聚类法中的重心法将凝聚点进行合并。

③ 将剩余的（$n-k$）个样本逐个引人，再计算它与已有 k 个凝聚点的距离，如最小距离$>r$，则该样本作为新的凝聚点；选定距离最小者，则将其并人最靠近该样本的凝聚点，再重新计算各类的重心，并以该重心作为新的凝聚点。重新验证新凝聚点两两之间的距离，如距离$<c$，则将相应的两凝聚点合并，用这两点的重心作为新凝聚点，重复此步骤，直至所有凝聚点之间的距离均大于等于 c。

④ 将 n 个样本再从头至尾逐个引人，用步骤③的办法归类，但与步骤③稍有不同。不同

之处在于：某个样本进入后，如分类与原来的一样，则重心不变；如分类与原来的不一样，则所涉及的两类重心需重新计算。如果 n 个样本通过后所分的类与原来的类没有改变，则过程停止，否则重复第④步。

由于上述 K-均值聚类法的分类结果仍有很大的不确定性，促使人们设法对它们进行改进，有关的研究也很多，其中以 Isodata 命名的方法是这一类方法中最精细的一个，它是美国标准研究所花了好几年的时间研究出来的[21]。此外，用全局最优化方法来改进 K-均值聚类法是目前化学计量学研究中的一个重要课题，将在后述章节加以适当介绍。

五、基于全局寻优的聚类法

【基本思路】

从前述讨论可知，聚类分析实际上可看成一个优化过程，它通过优化某一特定的目标函数而达到最好的分类。由于聚类目标函数的严重非线性，用传统的局部优化方法难以找到全局最优解，因此需用全局优化方法。对于 n 个样本划分为 c 类的问题，可能的组合划分数目为[26]：

$$s = \frac{1}{c!}\sum_{k=1}^{c}\left[(-1)^{c-k}\binom{c}{k}k^{n}\right] \tag{7-72}$$

若将 100 个样本分为两类，用穷举法需计算高达 $2^{99}-1$ 种划分，以确定最优解．且用其他确定性的全局优化方法也因计算量过大而难以实现。随机性的全局优化方法在处理上述问题上优于确定性全局优化方法，遗传算法和模拟退火作为随机性的全局优化方法已开始引入化学聚类分析[24,25]，它们的共同之处是将聚类分析构造成非连续的组合优化问题，即通过寻找各个样本对各类的划分而求得最优目标函数，由（7-42）式可知此类方法计算量巨大，需消耗很长时间才能找到最优点。在此讨论的是连续型优化模型，即通过找出各类中心的正确位置而达到正确聚类，使所有点到其所属类中心距离的和最小，避免了求解组合优化问题，从而减少了计算量，提高优化效率。遗传策略[27]被用于在连续的实数空间寻优，此法是一种借鉴生物界自然选择和自然遗传机制的随机搜索优化策略，并隐含并行性和对全局信息的有效利用能力是其两大特点，前者使遗传算法只须检测少量的结构就能反映搜索空间的大量区域，后者使遗传算法具有稳健性（即不会收敛至局部最优），故它尤其适于处理传统搜索方法解决不了的复杂和非线性问题（参见第四章）。在此，以遗传算法为例来说明基于全局寻优的聚类法，当然，其他全局最优方法，如模拟退火[28]和基于数论的序贯优化算法也同样是可行的，因它们的基本原理是类似的。

遗传算法通过模拟生物的遗传、变异和自然选择过程，将一代群体变换到新一代的群体。在这里每代群体由一组染色体组成，每条染色体代表搜索空间内的 1 个解。染色体由待求参数排列在一起构成。通过对上一代（父代）群体中的染色体进行有选择的复制、交叉和变异，可产生新一代（子代）群体，此过程一直重复，直至达到最优解。通用做法可简单描述如下：

① 染色体表示。给出搜索空间中染色体的表示形式。将待求参数的一组取值排列一处，构成一个串，此即一条染色体，它代表所求问题的一个可行解。

② 群体初始化。选择一个整数 n 作为群体的规模参数，然后在搜索空间内随机地任取 n 个点，每个点用一条染色体表示，这些染色体构成初始群体。

③ 计算目标函数值。将每条染色体上的参数取值代入所求的目标函数中，计算出对应的目标函数值 F。

④ 将目标函数值转换为适应性。若目标函数值不适于以后的交叉和变异过程，将其转换成适合的值。在我们讨论的问题中，需求目标函数的最小值，因此使目标函数值小的染色体适应性大，故采取如下转换：

$$\text{fittness}(i)=[\max(F)-F(i)]/[\max(F)-\min(F)] \tag{7-73}$$

⑤ 根据适应性大小，设计一个随机选择规则，复制出下一代群体。本文采用赌盘选择技术。此选择过程虽是随机的，但每个染色体被选择复制的机会却直接与其适应性成正比。

⑥ 交叉和变异。交叉是以概率 P_c 交换两条染色体间对应位置上的分量。变异是以概率 P_m 改变染色体上的每一位的值。本文通过如下方式实现：

$$x_{新}=x_{旧}\times(1\pm\text{rand}) \tag{7-74}$$

在此，rand 为(0,1)均匀分布的随机数。

整个优化过程通过第 3 步至第 6 之间的循环进行实现。目标函数收敛后停止。

在此介绍的方法是基于前述 K-均值聚类法的一种方法，首先确定各类中心，然后根据样本点与各中心的距离进行分类。因此构造的目标函数依赖于各类的中心。遗传算法用来搜索正确的类中心，若样本空间为 p 维，共分 c 个类，则需优化 $p*c$ 个实参数值。对于不同的方法，目标函数的构造过程不一样。可分别描述如下：

设样本数据矩阵可表示为

$$\boldsymbol{V}=\begin{bmatrix} v_{11}v_{12}\cdots v_{1p} \\ v_{21}v_{22}\cdots v_{2p} \\ \vdots \quad \vdots \quad \vdots \\ v_{j1}v_{j2}\cdots v_{jp} \\ \vdots \quad \vdots \quad \vdots \\ v_{n1}v_{n2}\cdots v_{np} \end{bmatrix}=\begin{bmatrix} \boldsymbol{v}_1^t \\ \boldsymbol{v}_2^t \\ \vdots \\ \boldsymbol{v}_j^t \\ \vdots \\ \boldsymbol{v}_n^t \end{bmatrix} \tag{7-75}$$

其中，n 为样本点数，p 为样本维数。各类中心可表示为

$$\boldsymbol{O}=\begin{bmatrix} o_{11}o_{12}\cdots o_{1p} \\ o_{21}o_{22}\cdots o_{2p} \\ \vdots \quad \vdots \quad \vdots \\ o_{i1}o_{i2}\cdots o_{ip} \\ \vdots \quad \vdots \quad \vdots \\ o_{c1}o_{c2}\cdots o_{cp} \end{bmatrix}=\begin{bmatrix} \boldsymbol{o}_1^t \\ \boldsymbol{o}_2^t \\ \vdots \\ \boldsymbol{o}_i^t \\ \vdots \\ \boldsymbol{o}_c^t \end{bmatrix} \tag{7-76}$$

c 为类数目，p 为样本向量的维数。优化的目标是使式（7-77）所示函数达到最小：

$$\text{minimize}F(\boldsymbol{W},\boldsymbol{O}) = \sum_{i=1}^{c}\sum_{j=1}^{n} w_{ij}D^2(o_i,v_j) \tag{7-77}$$

其中，$\boldsymbol{W}=\{w_{ij}\}$为 $c\times n$ 的权重矩阵；$\sum_{i=1}^{c} w_{ij}=1, w_{ij}\in\{0,1\}$。$D(\boldsymbol{v}_j,\boldsymbol{o}_i)$为第 j 个样本到第 i 个中心之间的欧氏距离，即

$$D(\boldsymbol{v}_j,\boldsymbol{o}_i)=\sqrt{\sum_{l=1}^{p}(v_{jl}-o_{il})^2} \tag{7-78}$$

式中，$D(\boldsymbol{v}_j,\boldsymbol{o}_i)$为第 j 个样本点到第 i 个中心之间的距离，且 $\min(v_{jl})\leqslant o_{il}\leqslant\max(v_{jl})(j=1,\cdots,n;i=1,\cdots,c;l=1,\cdots,p)$。若 $\boldsymbol{W}^*$、$\boldsymbol{O}^*$ 为最优的权重和类中心矩阵，对于上述问题可从两

个方面进行优化：一是寻找最优的 W^*，以求得最优的 $F(W^*, O)$ 值；二是寻找最优的 O^*，以求得最优的 $F(W, O^*)$ 值。前一方法为非连续的组合优化问题。后一方法为连续的优化问题，其实质是在 p 维的实数空间内搜索出 c 个点（最优类中心）。

【基本算法】

每条染色体代表 c 个类中心的一组取法，用一个实向量表示：$\boldsymbol{x}=(x_1, x_2, \cdots, x_d)$，其中 $d=p\times c$，p 为样本维数，c 为类的数目。

（1）将每条染色体向量转变为一个类中心矩阵，其每个元素的可由（7-79）式确定：

$$o_{ie}=x_{(i-1)\times p+e} \qquad (1\leqslant i\leqslant c, 1\leqslant e\leqslant p) \tag{7-79}$$

（2）根据（7-78）式计算样本 $\boldsymbol{v}_1^t$，$\boldsymbol{v}_2^t$，…，$\boldsymbol{v}_n^t$ 与各个中心 $\boldsymbol{o}_i^t$ 的欧氏距离，将每个样本划分至距离最近的类，得到权重 $\boldsymbol{W}$。

（3）根据 $\boldsymbol{W}$ 和 $\boldsymbol{V}$ 重新计算新的中心 $\boldsymbol{o}_1^t$，$\boldsymbol{o}_2^t$，…，$\boldsymbol{o}_c^t$：

$$\boldsymbol{o}_i=\frac{\sum_{j=1}^{n} w_{ij}\cdot \boldsymbol{v}_j}{\sum_{j=1}^{n} w_{ij}} \tag{7-80}$$

（4）根据（7-77）式计算目标函数值 F。

（5）将新的类中心转化为染色体，根据相应的 F 对其进行评价和遗传操作，然后转第 1 步，直至目标函数最小。

【讨论】

用本算法对两组分析数据进行了聚类分析：① 13 种牛黄样品中微量元素含量[29]，经标准化处理后的数据列于表 7-16；② 不同茶叶中几种有机物含量数据[30]。

表 7-16 经标准化处理后的 13 种牛黄样品中微量元素含量

试样编号①	Cr	Cu	Mn	Ti	Zn	Ph	Mo	Ca	K	Na	P
1	0.2601	1.2663	−0.3583	−0.8806	2.1462	0.7075	0.3092	2.4408	−0.8782	−0.7953	2.7953
2	−0.5501	−0.4793	0.4264	−0.7349	1.6575	0.9897	0.3092	1.1206	−0.9742	−0.8503	−0.7975
3	−0.2094	1.4416	1.3222	−0.9609	1.3797	0.2796	1.8715	0.7906	−0.7823	−0.7128	−0.9002
4	−0.1412	−0.7887	−0.3329	−0.9448	−0.4879	2.3681	−0.1121	0.8089	−0.3985	0.4146	0.3316
5	−0.0325	0.3886	0.9366	−0.8966	−0.3549	0.5646	−0.6738	−0.4288	0.7528	1.2395	−0.6790
6	0.4039	−0.1633	0.3890	−0.2495	−0.4589	−0.6124	−0.9546	−0.0439	1.3284	0.6620	−0.3356
7	−0.8455	1.6040	−0.8126	−0.2655	−0.5768	−0.0425	1.1693	−0.5755	−0.2066	0.3871	−0.6435
8	−0.5539	−0.9086	−0.7482	0.2371	−0.4448	−0.0360	0.5334	−0.8322	1.7122	1.9544	−0.4382
9	−0.5880	−0.6811	−0.4788	1.6784	−0.5340	−1.0765	−0.9546	−0.4471	1.0406	−0.7953	0.7422
10	−1.5648	−0.7790	−1.0007	−0.4273	−0.5804	−0.3776	−0.1121	−0.8505	0.8487	0.6620	−0.2329
11	0.0178	0.9968	−0.9148	0.6422	−0.5779	−0.4834	1.5906	−0.4838	0.3026	0.1671	0.7422
12	2.5159	−0.8352	−0.6767	1.9193	−0.5841	−1.1460	0.9546	−0.6580	−1.1660	−1.4827	−0.7462
13	1.4905	−1.0622	2.2487	0.8831	−0.5835	−1.1406	−0.9546	−0.8413	−0.9742	0.8503	0.4382

① 试样：1～3 为自然生成牛黄；4～13 为人工培养牛黄。

在进行聚类分析采用的遗传算法中，父代群体规模为 20，通过复制、交叉和变异后产生 80 条子代染色体，再从中选出 20 条好的，作为新的父代。其中交叉概率为 0.7，变异概率为 0.2。对各套数据分类结果讨论如下：

牛黄是一种常用中药，分为天然的和人工的。在表 7-16 列出的 13 个样本中，1～3 号为天然牛黄，4～13 号为人工牛黄。对此数据采用遗传法计算，均能在十代之内得出最优结果：

1～4 号为第一类，5～13 号为第二类。最优目标函数值为 $F^*=94.2989$，优于文献［24］用模拟退火求得的结果（$F^*=94.3589$，第 7 号样本被误分至第一类）。值得提出的是，若将第 4 号按原数据分类划为第二类时，目标函数值将增大至 95.2626，似不太合理。

茶叶是一种消耗量很大的商品，由于产地、加工方式等不同，故种类较多，质量不一，Liu 等人[30]曾分析了 31 种茶叶，其中包括三类：绿茶、黑茶和乌龙茶，每类中又各含两种。根据茶叶专家品偿结果将各种茶标号，最好的计为 1 号。对此分析数据采用谱系聚类方法进行了分类。他们的结果可概括为

第一类：C_1～C_4，H_1～H_3，K_1～K_2，F_1～F_4

第二类：C_5～C_7，H_4～H_5，K_3～K_4，F_5～F_7

第三类：T_1～T_4，S_1～S_4

根据此结果，可求得目标函数值 F 为 50.8600。

采用本文所述分类方法结果为

第一类：C_1～C_4，H_1～H_3，K_1，F_1～F_4

第二类：C_5～C_7，H_4～H_5，K_2～K_4，F_5～F_7

第三类：T_1～T_4，S_1～S_4

得目标函数值 F 为 50.6999。结果与文献［24］中一致，且最长可在 10min 以内收敛至最优点，从速度上看明显优于文献［24］中的模拟退火方法（需 68min）。

对上述两类实际体系的分析表明，用遗传算法进行聚类分析，在连续的实数空间中寻找最优的各类中心，计算量比非连续的组合优化小，分类合理，是一种有效的无监督模式识别方法。

六、模糊聚类法

在聚类问题上可采用硬分类和模糊分类两种方法确定目标函数。所谓硬分类的结果是每个样本将绝对地划归某一类，即每个样本仅能属于一个类；而模糊分类的结果是每个样本可在不同程度上或多或少地属于各个不同的类。文献[24,25］中所用分类方法均属硬分类。模糊分类方法在分析化学文献中少见，HoriaF. Pop 等人[26]最近在这方面作了有益尝试，但他采用的是用局部优化方法进行谱系聚类。在此，我们将介绍用遗传算法来进行模糊分类中的方法。

【基本思路】

模糊分类是根据 Bezdek[33]提出的方法，如仍采用前节所定义的符号，则模糊分类与前节不同之点仅在于权重阵 $\boldsymbol{W}$，对于硬分类权重阵 $\boldsymbol{W}$ 的元素只能取 1 或 0，而模糊分类中的权重阵 $\boldsymbol{W}$ 的取值可在（0，1）连续变化。它也是从初始模糊权重阵 $\boldsymbol{W}$ 或类中心阵 $\boldsymbol{O}$ 开始，用下述两式进行迭代，

$$\boldsymbol{o}_i=\frac{\sum_{j=1}^{n}w_{ij}^2 * \boldsymbol{v}_j}{\sum_{j=1}^{n}w_{ij}^2} \tag{7-81}$$

$$w_{ij}=\frac{1}{\sum_{l=1}^{c}\left(\frac{D(\boldsymbol{v}_j,\boldsymbol{o}_i)}{D(\boldsymbol{v}_j,\boldsymbol{o}_l)}\right)^2} \tag{7-82}$$

以上方法称为模糊 c 均值方法（FCM）。寻找最优的 $\boldsymbol{O}^*$ 或 $\boldsymbol{W}^*$ 都是连续的优化问题。找到 $\boldsymbol{O}^*$ 或 $\boldsymbol{W}^*$ 中的一个，即可利用上述两式来相互求另一个。若先求 $\boldsymbol{W}^*$，需计算 $c\times n$ 个参数，若

先求 O^*，需计算 $c\times p$ 个参数。因一般有 $p<n$，故求 O^* 计算量小些。因此本文是求最优的类中心（O^*）以使目标函数达到最小。直至达到某一收敛准则。（如 $\| W^{(k)}-W^{(k-1)} \| <\varepsilon$）。

【基本算法】

具体算法可表示如下：

目标函数为 $$\text{minimize}F(\boldsymbol{W},\boldsymbol{O}) = \sum_{i=1}^{c}\sum_{j=1}^{n} w_{ij}^2 D^2(\boldsymbol{o}_i,\boldsymbol{v}_j) \tag{7-83}$$

W 为 $c\times n$ 的模糊权重阵。其中 $\sum_{i=1}^{c} w_{ij} = 1, w_{ij}\in[0,1]$。在此权重可在［0，1］区间连续取值，而硬分类权重只可取 0 或 1。同样利用式 $D(\boldsymbol{v}_j,\boldsymbol{o}_i)=\sqrt{\sum_{l=1}^{p}(\boldsymbol{v}_{jl}-\boldsymbol{o}_{il})^2}$ 计算距离 $D(\boldsymbol{v}_j,\boldsymbol{o}_i)$，继用下述（7-84）与（7-85）两式计算新的类中心和权重：

$$\boldsymbol{o}_i = \frac{\sum_{j=1}^{n} w_{ij}^2 * \boldsymbol{v}_j}{\sum_{j=1}^{n} w_{ij}^2} \tag{7-84}$$

$$\boldsymbol{w}_{ij} = \frac{1}{\sum_{r=1}^{c}\left(\frac{D(\boldsymbol{v}_j,\boldsymbol{o}_i)}{D(\boldsymbol{v}_j,\boldsymbol{o}_r)}\right)^2} \tag{7-85}$$

寻找最优的 O^* 或 W^* 都是连续的优化问题。找到 O^* 或 W^* 中的一个，即可利用其中一个来求另一个。若先求 W^*，需计算 $c\times n$ 个参数，若先求 O^*，需计算 $c\times p$ 个参数。因一般有 $p<n$，故求 O^* 计算量小些。遗传算法用来优化 F 的过程基本与硬分类同，只是计算权重、类中心是利用上述（7-84）与（7-85）两式。

【讨论】

用本算法对两组分析数据进行了聚类分析：① 两类烟叶化学成分分析结果[31]；② 古罗马陶器碎片中几种主要和次要无机成分含量数据（表 7-17）[32]。

烟草的品种与其化学成分之间的关系密切，由于化学成分数目较多，在此采用了 26 种烟的 16 种主要成分。其中第 1～13 号为晒晾烟，第 14～26 号为烤烟。用本文所述模糊聚类分类方法分析，结果为第 2～13 号为一类，第 1、第 14～26 号为第二类，目标函数值为 8.8930×10^3。第 1 号对第一类的权重为 0.2987，对第二类的为 0.7013，若将其分别改为 0.51 和 0.49，目标函数将上升至 8.9732×10^3，可见第 1 号应分至第二类。

表 7-17　古罗马陶器碎片中几种主要和次要无机成分含量数据

试　样	K_2O	MgO	CaO	TiO_2	MnO	Fe_2O_3	Al_2O_3
02A	2.08	3.55	8.63	0.77	0.08	6.16	17.40
02B	1.88	3.33	9.52	0.93	0.13	7.07	18.20
02C	2.08	3.40	8.31	0.91	0.14	7.03	19.80
02D	1.94	3.17	9.99	0.95	0.15	7.32	18.80
03	2.54	1.37	11.50	0.82	0.11	5.18	23.10
05A	3.15	8.00	1.85	0.92	0.14	8.88	19.10
05B	3.13	8.45	1.81	0.91	0.14	9.12	20.00
05C	2.77	7.67	1.71	0.89	0.11	8.84	19.10
05E	2.96	7.80	1.69	0.88	0.12	9.24	19.60
05H	1.72	1.45	1.02	1.40	0.01	6.58	18.60
05I	3.04	1.84	3.31	1.02	0.09	7.14	20.80

续表

试 样	K_2O	MgO	CaO	TiO_2	MnO	Fe_2O_3	Al_2O_3
06A	3.06	4.79	1.48	0.84	0.18	9.32	21.90
06B	3.03	2.00	11.60	1.18	0.06	5.90	23.30
06C	1.98	3.51	8.98	0.91	0.16	7.30	18.40
08	2.61	3.08	6.14	1.01	0.10	7.92	18.80
09A	1.96	3.57	10.10	0.89	0.17	7.48	18.30
09B	3.61	4.63	3.94	0.84	0.11	8.60	19.80
09C	2.24	4.09	8.70	0.86	0.09	6.81	17.40
10A	1.93	1.38	0.75	1.7[illegible]	0.02	8.32	25.50
10B	1.88	1.41	0.76	1.72	0.02	8.21	25.10
11A	2.75	1.34	8.23	0.84	0.07	5.67	20.80
11B	2.86	2.10	10.90	1.28	0.06	6.05	23.30
12A	2.26	1.27	10.60	0.93	0.03	4.97	19.20
12B	2.88	2.02	11.50	1.10	0.07	6.33	22.20
13A	3.60	35.32	1.78	0.96	0.15	8.93	19.80
13B	1.76	3.26	12.30	0.86	0.14	6.01	17.30
13C	2.18	3.40	9.73	0.81	0.08	6.19	16.00
13D	2.10	3.37	9.00	0.94	0.16	7.39	17.40
13E	1.92	3.22	12.50	0.93	0.15	7.76	17.00
13F	2.02	3.21	11.10	0.93	0.14	7.18	17.30
14C	2.92	1.19	12.60	0.85	0.07	5.37	21.80
14E	3.20	1.94	9.71	1.21	0.05	5.76	22.30
15A	3.19	2.01	12.30	1.32	0.07	6.30	21.70
15B	3.06	1.78	9.53	1.20	0.05	6.62	20.90
16A	3.18	8.08	1.73	0.89	0.12	9.22	19.20
16B	2.97	8.59	1.78	0.86	0.11	8.90	18.90
16C	3.08	3.97	1.40	0.90	0.09	6.86	19.50
16D	3.09	6.20	1.81	0.93	0.15	9.39	20.30
16E	2.68	5.78	2.25	1.03	0.12	8.60	19.00
16F	2.49	8.12	2.18	0.88	0.09	8.32	18.00
16G	2.85	8.00	2.15	0.88	0.14	8.91	19.10
16H	3.25	7.43	1.65	0.89	0.11	9.05	19.80
16I	3.19	5.95	1.86	0.93	0.13	8.47	20.10
22	3.33	4.90	1.12	0.88	0.17	10.50	23.60
25	3.10	3.01	0.99	0.93	0.14	8.30	22.00
27	2.04	1.37	0.81	1.66	0.02	8.80	24.40
28	2.95	6.63	1.82	0.96	0.14	9.90	21.50
32	3.19	6.93	1.61	1.03	0.10	8.70	21.50

古罗马陶器碎片中几种主要和次要无机成分含量数据是用原子吸收和发射方法对 48 种古代陶器中 7 种主要和次要成分进行分析的结果。对其分类既是化学上的问题也是考古学的问题。由于遗传算法可在高维变量空间进行快速、有效地搜索，故直接将原始数据标准化后作模糊分类，不进行降维，以充分保留原有数据结构。经典的模糊 c 均值方法（FCM）易于陷入鞍点或局部最优点[34]，我们引入遗传算法分类，算法收敛情况见图 7-9，从图上看开始目标函数下降较快，接近最优点附近以后下降较慢。用本方法求得的目标函数值为 $F=37.9785$，而用经典的 FCM 方法任取初始点计算 5 次，求得的目标函数值各不相同，分别为 46.8121、46.8050、37.9785、46.8050、46.8050，可见经典 FCM 方法在此确实易陷入局部最优点。而本算法不受影响，在各次计算中最终结果是确定的，均为 37.9785。从这点上可以反证用本方

法找到的可能是全局最优。

如沿用文献［32］的样本编号，分类的结果具体如下：

第一类：02A，02B，02C，02D，06C，08，09A，09C，13B，13C，13D，13E，13F

第二类：06A，09B，13A，16C，16D，16I，22，25，28，32

第三类：03，05I，06B，11A，11B，12A，12B，14C，14E，15A，15B

第四类：05A，05B，05C，05E，16A，16B，16E，16F，16G，16H

第五类：05H，10A，10B，27

第一类与文献［32］一致，即原文中的D类。第二类由文献中未能分类的零散点构成。为便于考查零散点与各类的关系，将第二类所有点的权重列于表7-18。值得提出的是，文献［32］中，此类的06A和09B被误分为E类（与本文第四类相对应）。该文作者亦知从考古学家的观点看它们不应属E类。我们的方法在这里成功地将其从E类中剔出。第三类与文献［32］的结果基本一致。在文献［32］中它们先被分为B类和C类，然而，在讨论中结合考古学知识分析时，发现将它们分为一类似更合理，这与本法的结果不谋而合。另外，原来未被分类的样本05I也被分入第三类，其权重见表7-18。如从权重看，05I确为介乎第二类（0.2998）和第三类（0.3237）之间的点，故此实际上是与文献［32］的结果相吻合的。第四类与文献［32］中的E类大致相对应。不同的是本方法将样本16E和16F（文献［32］中它们未被分类）归入此类。有趣的是，在文献［32］的讨论部分中，作者按考古学知识得出结论，这些点应该分入E类（即本文第四类），但用他们的方法未能达到。第五类与文献［32］结果（即该文中的A类）也基本一致。只是本文将该文中未被归类的05H也分入此类，从它对各类的权重看，05H是介乎第三类（0.2275）和第五（0.3142）类之间的点。

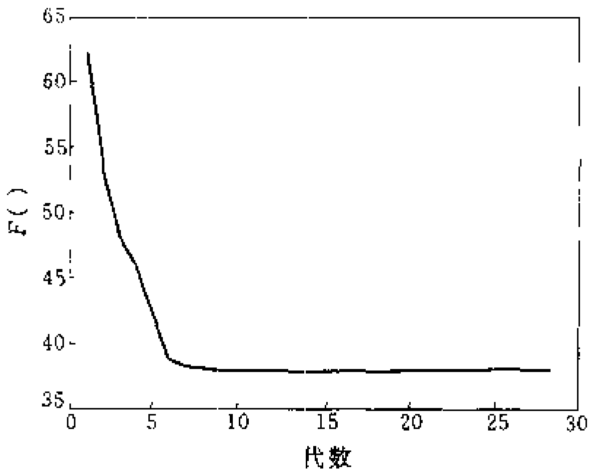


图7-9　古罗马陶器碎片数据的模糊聚类算法收敛情况

所有与文献［32］中分法不同的点对各类的权重见表7-18。

总之，本方法分类能在高维空间中进行有效寻优，分类结果似比文献［32］更合理。另外，从所得模糊权重再仔细查看样本点与各类的精细关系，可能更有利于对所得结果进行进一步的探讨研究。

表7-18　与文献［32］中分法不同的点对各类的权重

样　本	每　类　的　权　重				
	1类	2类	3类	4类	5类
06A	0.0735	0.6023	0.0534	0.2403	0.0305
09B	0.0662	0.5702	0.0705	0.2680	0.0251
13A	0.0433	0.6478	0.0381	0.2523	0.0185
16C	0.1446	0.3834	0.1635	0.2586	0.0499
16D	0.0283	0.5869	0.0196	0.3541	0.0111
16I	0.0168	0.7383	0.0134	0.2254	0.0062
22	0.0894	0.4940	0.0845	0.2731	0.0590

续表

样 本	每 类 的 权 重				
	1类	2类	3类	4类	5类
25	0.0723	0.6544	0.0765	0.1612	0.0356
28	0.0467	0.4864	0.0561	0.4065	0.0242
32	0.0454	0.4524	0.0514	0.4220	0.0288
05H	0.1909	0.1375	0.2275	0.1298	0.3142
05I	0.1500	0.2998	0.3237	0.1554	0.0711
08	0.3382	0.2549	0.1738	0.1835	0.0497
16E	0.0868	0.3634	0.0173	0.4768	0.0255
16F	0.1227	0.2094	0.0642	0.5676	0.0361

第六节　特征投影的模式识别方法

对于含有 d 个参量的模式矢量，是 d 维空间中的一个点，如 d 等于 2 或 3，则可直接用图形显示这些模式矢量。图形显示具有直观性，现代计算机提供了强有力的屏幕图形显示功能，而人眼具有很强的识别图形能力，同样的字由不同的人书写，人眼能识别其细微差别，辨识手迹真伪。在二维或三维空间中显示的模式分布情况，借人眼常能识别存在的分类图景，也就是说人眼是强有力的模式识别器。而当 $d>3$ 时，这种直接显示则不可能，这就涉及降维的问题。如何将 d 维空间的图景，在二维或三维空间中显示出来，并尽可能减少原 d 维空间中分类信息的丢失？这就是将在这一节要讨论的基于投影的模式识别方法主要思路。

基于投影的模式识别方法也有很多，其中最重要包括主成分分析的投影判别法（亦称 Karhunen-Loeve 变换，K-L 分解法）、SIMCA 方法、基于偏最小二乘分解的特征投影法、非线性映照投影法等，因其具有图形显示具有直观性，很容易为化学家所理解，故它们在化学模式识别中得到相当广泛的应用[35~37]。本节将对这些方法作出介绍。

一、基于主成分分析的投影判别法

基于主成分分析的投影判别法采用多元统计中的主成分分析方法，先对样本量测矩阵 X 直接进行分解，只取其中的主成分来投影，然后进行判别分析，故有主成分分析的投影判别法之称。对样本矩阵进行直接分解在数学上有几种方法。在化学计量学中一般采用的方法是非线性迭代偏最小二乘算法（NIPALS）[38]。这种方法实际上是沿用 Von Mises 的乘幂法[39]。另一种方法是线性代数中常用的奇异值分解法（SVD）。奇异值分解法可将任意阶实数分解成为三个矩阵的积，即

$$\boldsymbol{X}=\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{t}$$

式中，$\boldsymbol{S}$ 为对角矩阵，它收集了 $\boldsymbol{X}$ 矩阵的特征值，$\boldsymbol{U}$ 和 $\boldsymbol{V}^{t}$ 分别为标准列正交和标准行正交矩阵，收集了这些特征值所对应的列特征矢量和行特征矢量，在多元统计的主成分分析中，一般被称为得分矩阵和荷载矩阵。图 7-10 示出了主成分分析的数学与几何意义。一般只需取前几个大特征值所对应的特征矢量作为主成分，样本量测矩阵 X 的样本矢量直接对它们投影就可得到所需的主成分分析特征投影图（参见图 7-10）。

【基本思路】

主成分分析的投影判别法的基本思路是基于主成分分析所得的主成分轴是该数据矩阵的最大方差方向，且这些主成分轴相互正交（参见第六章第四节），这样，就可保证在从高维向

低维空间投影时尽量多地保留有用信息。如果对样本量测数据矩阵 X 的构成作如下规定，我们可以很清楚地看出主成分分析的投影性质。令样本数据矩阵可表示为

$$\boldsymbol{X}=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & & \vdots \\ x_{j1} & x_{j2} & \cdots & x_{jd} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}=\begin{bmatrix} \boldsymbol{x}_1^t \\ \boldsymbol{x}_2^t \\ \vdots \\ \boldsymbol{x}_j^t \\ \vdots \\ \boldsymbol{x}_n^t \end{bmatrix}=[\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \cdots \boldsymbol{x}_d]$$

式中，n 为样本点数，d 为样本维数。$\boldsymbol{x}_i^t$（$i=1$，2，…，n）称为样本向量，$\boldsymbol{x}_j$（$j=1$，2，…，d）称为变量向量。因为

$$\boldsymbol{X}=\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^t \qquad \text{即} \qquad \boldsymbol{X}\boldsymbol{V}=\boldsymbol{U}\boldsymbol{S}$$

亦即

$$\boldsymbol{X}\boldsymbol{V}=\begin{bmatrix} \boldsymbol{x}_1^t \\ \boldsymbol{x}_2^t \\ \vdots \\ \boldsymbol{x}_n^t \end{bmatrix}\cdot[\boldsymbol{v}_1,\ \boldsymbol{v}_2,\ \cdots,\ \boldsymbol{v}_A]=\begin{bmatrix} \boldsymbol{x}_1^t\boldsymbol{v}_1 & \boldsymbol{x}_1^t\boldsymbol{v}_2\cdots\boldsymbol{x}_n^t\boldsymbol{v}_A \\ \boldsymbol{x}_2^t\boldsymbol{v}_1 & \boldsymbol{x}_2^t\boldsymbol{v}_2\cdots\boldsymbol{x}_n^t\boldsymbol{v}_A \\ \vdots & \vdots \quad\quad \vdots \\ \boldsymbol{x}_n^t\boldsymbol{v}_1 & \boldsymbol{x}_2^t\boldsymbol{v}_2\cdots\boldsymbol{x}_n^t\boldsymbol{v}_A \end{bmatrix}=\boldsymbol{U}\boldsymbol{S}$$

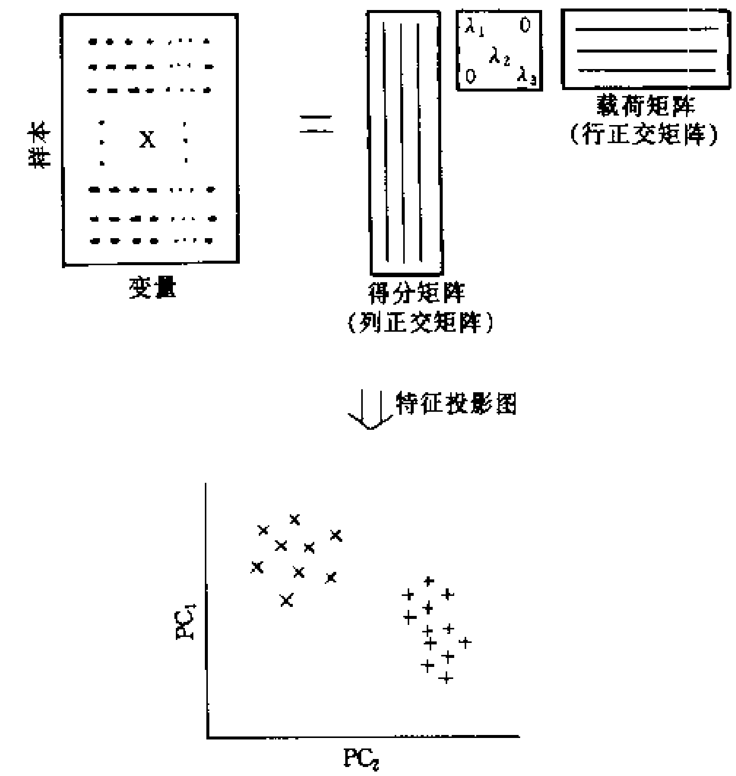


图 7-10 主成分分析的数学与几何意义示意图

可见矩阵 $\boldsymbol{U}\boldsymbol{S}=\boldsymbol{T}$ [亦称非标准化的得分（Scores）矩阵] 的每一个元素实际是每一个样本向量 $\boldsymbol{x}_i^t$（$i=1$，2，…，n）对荷载（loadings）矩阵 $\boldsymbol{V}$ 中的每一相互正交的荷载矢量上的投影坐标（内积本质上就是投影），它反映了样本与样本之间的相互关系；同理可得，荷载（loadings）矩阵的每一个元素实际是每一个变量向量 $\boldsymbol{x}_j$（$j=1$，2，…，d）对得分矩阵中的每

一相互正交的得分矢量上的投影坐标，它反映了变量与变量之间的相互关系。

另外，原始的主成分分析一般是对数据协方差阵进行正交变换而得（参见第六章第四节），即

$$Q^tX^tXQ=\mathrm{diag}[\lambda_1,\lambda_2,\cdots,\lambda_d]$$

式中，Q 为正交变换矩阵，λ_1，λ_2，…，λ_d 称为数据协方差阵 X^tX 的特征值。它与前述的奇异值分解法所得的 3 个分解矩阵，即对角矩阵 S，标准列正交矩阵 U 和标准行正交矩阵 V^t 分别有什么关系呢？

因为

$$X=USV^t$$

即

$$X^tX=VSU^tUSV^t=VSSV^t=\mathrm{diag}[\lambda_1,\lambda_2,\cdots,\lambda_d]=Q^tX^tXQ$$

又因为 $SS=\mathrm{diag}[s_1,s_2,\cdots,s_d]\times\mathrm{diag}[s_1,s_2,\cdots,s_d]=\mathrm{diag}[s_1^2,s_2^2,\cdots,s_d^2]$，而且，$X^tX=VSSV^t$，即 $V^tX^tXV=\mathrm{diag}[s_1^2,s_2^2,\cdots,s_d^2]$，还因为 V^t 为标准行正交矩阵，$V^tV=I$，所以有

$$V=Q \qquad s_i^2=\lambda_i \qquad (i=1,2,3,\cdots,d)$$

这就说明了经典主成分分析与奇异值分解法的关系。在 MatLab 程序中，做主成分分析就完全可以用奇异值分解法来替代了，实因在 MatLab 程序中奇异值分解法特别简单，就用下述一句话即可得所需结果：

$$[U,S,V]=\mathrm{svd}(X)$$

【基本算法】

主成分投影判别法的算法十分简单，它可由下述步骤所组成：

(1) 对前述的样本数据矩阵 X 进行奇异值分解，得

$$X=USV^t$$

(2) 令 $US=T=[t_1, t_2, \cdots, t_d]$，得非标准化的得分（scores）矩阵，一般取 $\sum s_i$ 等于 85%总方差所对应的得分矢量 $t_1,t_2,\cdots,t_A$ 来进行作图，即用 $t_i(i=1,2,\cdots,A)$ 对 $t_j(j=1,2,\cdots,A)$ 作图，得如图 7-11 所示的主成分特征投影图，不同类的样本点聚集在不同的位置。一般说来，前两个对应于最大特征值的得分矢量所得的投影图所含信息最多。

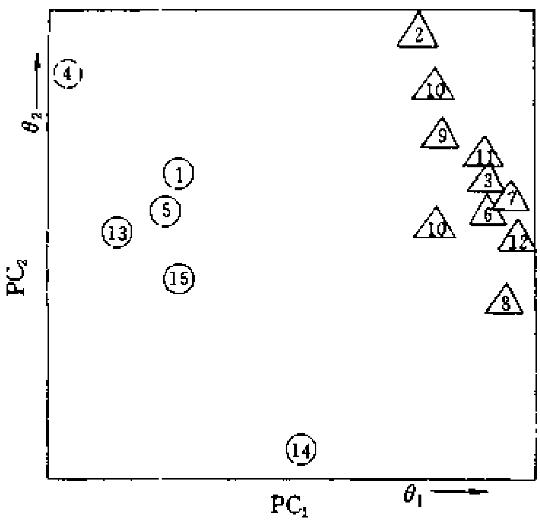


图 7-11　脑组织试样色谱数据的主成分投影图

基于主成分分析的投影判别法既可用于判别又可用于聚类，因我们可很容易地从投影图中看出样本与样本的关系来，故主成分分析的投影判别法在化学计量学中得到了非常广泛的应用。

Jellum[40]等用毛细管色谱法对 16 个脑组织试样进行分析，在色谱图中取多达 156 参量（可辨认的 156 个峰处的峰高），组成（16×156）阶矩阵，通过将矩阵作主成分分解，分别求得对应于两个最大特征值的得分矢量 t_1 和 t_2，并以 t_1 和 t_2 为投影轴作图，得到如图 7-11 的显示图形。三角形是有肿瘤的脑组织样，圆是正常脑组织样。有趣的是，4 号样原标签为“肿瘤脑组

织”样，但从显示图看，这个样本应属正常脑组织样、经查核，4 号样确属正常脑组织误标为“肿瘤样”。从这个例子可以看出，用主成分分析的方法，取最大的 2（或 3）个特征值所对应的得分矢量，可在二维空间（或三维空间，多数微型计算机作图软件有三维作图功能）显示样本在模式空间的分布，用人眼即能进行分类和判别。

二、基于主成分分析的 SIMCA 分类法

本节将介绍化学模式识别中一个较通用的方法——SIMCA 法[41~43]，这一方法根据其设计者定名的含义[42]是“统计匀线性多元分析”，或“簇类的独立软模式”，意为各类样本独立地建立局部“软”模式，或简易分类算法。第二个解释较确切地反映了此法的特点。

【基本思路】

SIMCA 方法是一种建立在主成分分析基础上的一种模式识别方法，其基本思路是先利用主成分分析的显示结果得到一个样本分类基本印象，然后分别对各类样本建立相应的类模型，继用这些类模型来对未知样品进行判别分析，以确定其属于哪一类，或不属于哪一类。

【基本计算过程】

在介绍 SIMCA 如何具体建立类模型之前，首先介绍它的计算全过程，以期读者先对该方法的全貌有一个完整的印象。

SIMCA 方法的计算全过程由图 7-12 示出的框图给出。从此框图可以看出，SIMCA 实际上是在循环地使用主成分分析方法，它先是用主成分分析方法来得到整个样本的分类，然后，在此基础上，分别建立各类样本的主成分模型，继用它们来检验未知样本，判别未知样本的类别。由于整个 SIMCA 计算过程可在投影图上直接进行，所以，只要计算机程序的人机对话界面建立得好，此方法可完全在图上直接操作完成，使用起来非常方便。将 SIMCA 编人商业软件的例子很多，比较著名的有 Unscramber 和 Serius 的软件。

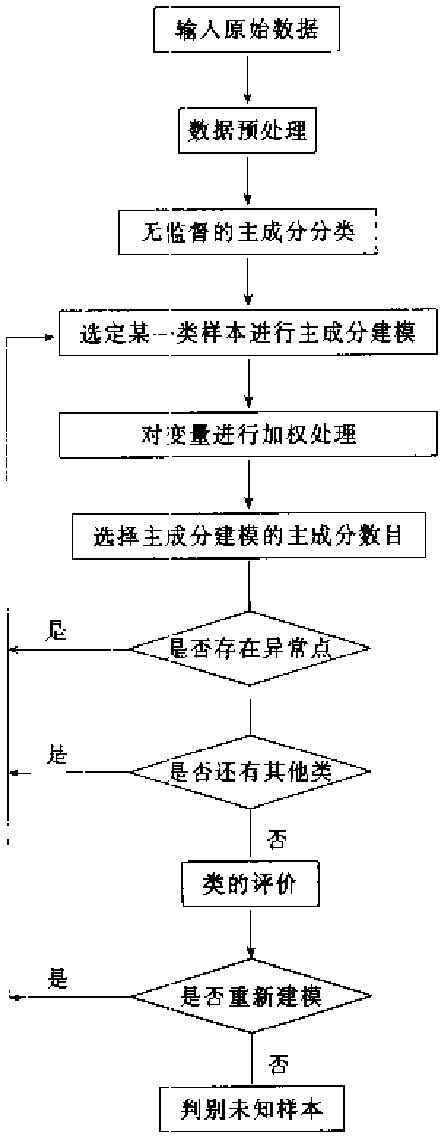


图 7-12　SIMCA 计算过程框图

从上述讨论可知，在整个 SIMCA 方法的计算过程中，其与基于主成分分析投影判别不同的地方在于它对每一类样本进行了主成分建模，在此将对这一部分进行较详细的讨论。为讨论方便，首先介绍基于化学计量学文献习惯用法的一些表达式。首先，令样本数据矩阵表示为

$$\boldsymbol{X}=\begin{bmatrix} x_{11}x_{12}\cdots x_{1d} \\ x_{21}x_{22}\cdots x_{2d} \\ \vdots\quad\vdots\quad\vdots \\ x_{j1}x_{j2}\cdots x_{jd} \\ \vdots\quad\vdots\quad\vdots \\ x_{n1}x_{n2}\cdots x_{nd} \end{bmatrix}=\begin{bmatrix} \boldsymbol{x}_1^t \\ \boldsymbol{x}_2^t \\ \vdots \\ \boldsymbol{x}_j^t \\ \vdots \\ \boldsymbol{x}_n^t \end{bmatrix}$$

式中，n 为样本点数，d 为样本维数。x_i^t（$i=1,2,\cdots,n$）称为样本向量，则一个样本就由样本矩阵中的一个行向量来表示。又因为

$$X=lx_c^t+USV^t+E=lx_c^t+TV^t+E \tag{7-86}$$

式中，$US=T$，T 矩阵亦称非标准化的得分（scores）矩阵；$l=[1,1,\cdots,1]^t$，为一元素全为1的矢量，其维数为 n，等于样本的数目；x_c 为所有样本的中心矢量。

式（7-86）说明，在进行主成分分析之前，对样本数据矩阵进行了中心化预处理。对于每一个样本矢量，根据式（7-86）有

$$x_i^t=x_c^t+\sum_k t_{ik}v_k^t+e_i^t \qquad (i=1,2,\cdots,n) \tag{7-87}$$

在上面两式中我们有意还加入了误差矩阵 E，所以在第二个式中也引入了一个误差矢量 e_i^t 项。引入误差项应该是不难理解的，实因在主成分分析中，一般都只取了对应于几个大特征值的特征向量作为主成分，那些很小或只占百分之几的特征值所代表的特征向量都包括在误差向量之中了。另外，由（7-86）式变为（7-87）式也是不难理解的，如果令 $X'=X-lx_c^t$，并略去（7-87）式中的误差项，就可以通过简单的代数运算而得到上述的样本矢量表达式。因

$$X'=\begin{bmatrix} x'_{11}x'_{12}\cdots x'_{1d} \\ x'_{21}x'_{22}\cdots x'_{2d} \\ \vdots \quad\quad \vdots \\ x'_{j1}x'_{j2}\cdots x'_{jd} \\ \vdots \quad\quad \vdots \\ x'_{n1}x'_{n2}\cdots x'_{nd} \end{bmatrix}=\begin{bmatrix} x_1'^t \\ x_2'^t \\ \vdots \\ x_j'^t \\ \vdots \\ x_n'^t \end{bmatrix}=TV^t=\begin{bmatrix} t_{11}t_{12}\cdots t_{1A} \\ t_{21}t_{22}\cdots t_{11} \\ \vdots \quad\quad \vdots \\ t_{j1}t_{j2}\cdots t_{jA} \\ \vdots \quad\quad \vdots \\ t_{n1}t_{n2}\cdots t_{nA} \end{bmatrix}V^t$$

$$=\begin{bmatrix} t_{11}t_{12}\cdots t_{1A} \\ t_{21}t_{22}\cdots t_{2A} \\ \vdots \quad \vdots \quad \vdots \\ t_{j1}t_{j2}\cdots t_{jA} \\ \vdots \quad \vdots \quad \vdots \\ t_{n1}t_{n2}\cdots t_{nA} \end{bmatrix}\begin{bmatrix} v_{11}v_{12}\cdots v_{1d} \\ \vdots \quad \vdots \quad \vdots \\ v_{j1}v_{j2}\cdots v_{jd} \\ \vdots \quad \vdots \quad \vdots \\ v_{A1}v_{A2}\cdots v_{Ad} \end{bmatrix}=\begin{bmatrix} \sum_k t_{1k}v_{k1} \sum_k t_{1k}v_{k2}\cdots \sum_k t_{1k}v_{kA} \\ \sum_k t_{2k}v_{k1} \sum_k t_{2k}v_{k2}\cdots \sum_k t_{2k}v_{kA} \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ \sum_k t_{jk}v_{k1} \sum_k t_{jk}v_{k2}\cdots \sum_k t_{jk}v_{kA} \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ \sum_k t_{jk}v_{k1} \sum_k t_{jk}v_{k2}\cdots \sum_k t_{jk}v_{kA} \end{bmatrix}$$

$$=\begin{bmatrix} \sum_k t_{1k}\ [v_{k1}v_{k2}\cdots v_{kA}] \\ \sum_k t_{2k}\ [v_{k1}v_{k2}\cdots v_{kA}] \\ \vdots \quad\quad\quad \vdots \\ \sum_k t_{jk}\ [v_{k1}v_{k2}\cdots v_{kA}] \\ \vdots \quad\quad\quad \vdots \\ \sum_k t_{nk}\ [v_{k1}v_{k2}\cdots v_{kA}] \end{bmatrix}=\begin{bmatrix} \sum_k t_{1k}U_k^t \\ \sum_k t_{2k}U_k^t \\ \vdots \quad \vdots \\ \sum_k t_{jk}U_k^t \\ \vdots \quad \vdots \\ \sum_k t_{nk}U_k^t \end{bmatrix} \tag{7-88}$$

这正是前述的样本矢量表达式 $x_i'^t=\sum_k t_{ik}U_k^t+e_i^t$，亦即 $x_i^t=x_c^t+\sum_k t_{ik}U_k^t+e_i^t(i=1,2,\cdots,n)$。有了这样一些说明，讨论主成分分析建模就不困难了。从（7-88）式子可以看出，每一个样本矢量可以表为荷载矢量的线性组合与误差矢量之和，而且，由于荷载矢量是相互正交的，这为主成分分析建模的计算带来了很大方便。

【主成分建模算法】

（1）定义样本到主成分模型的距离

$$s_i = e_i^t e_i / (d - A) \tag{7-89}$$

这是因为样本矢量表达式的前一部分，即 $t_{ik}v_k^t$ 正是主成分荷载矢量的线性组合，属于可由主成分模型表出的部分，在样本矢量表达式中没被主成分模型所包含的就是误差矢量，所以，样本到主成分模型的距离就是误差矢量的点积，(7-89) 式中的分母是样本矢量在建立了主成分模型，即降维后的自由度，由原样本空间的维数减去主成分数而得。s_i 在 SIMCA 方法中常被称为第 i 个样本的残余标准偏差（residual standard deviation，*RSD*)。

(2) 定义整个类的均值标准偏差 s_c

如令 $s = [s_1, s_2, \cdots, s_i, \cdots, s_m]^t$，则整个类的均值标准偏差 s_c 可由式 (7-90) 给出

$$s_c = s^t s / (n - A - 1) \tag{7-90}$$

(7-90) 式以 $(n-A-1)$ 为分母的理由是为了整个类的均值标准偏差不受该类内样本数多少的影响，而且，还考虑了由于引人 A 个主成分所失去的自由度。

(3) 定义 F 检验统计量

为了判别某一样本与某一类主成分模型的差别或相似程度，SIMCA 方法的提出者 Wold 引入了 F-检验统计量。从上面的讨论可知，某一样本的残余标准偏差的平方 s_i^2 和整个类的均值标准偏差的平方 s_c^2 分别具有自由度$(d-A)$和$(d-A)(n-A-1)$，据此，我们可以采用自由度为$(d-A)$和$(d-A)(n-A-1)$的 F-检验来判别某一样本与某一类主成分模型的差别或相似程度。F-检验（或记为 F_{crit}）可用来求得一个判定某样本是否属于某类的残余标准偏差平方的上界值：

$$s_{max}^2 = s_c^2 F_{crit}$$

用于 F-检验的置信水平一般取 0.05 或 0.01 即可。图 7-13 给出了置信水平取 0.05 或 0.01 的由一个主成分建立的类模型的例子。

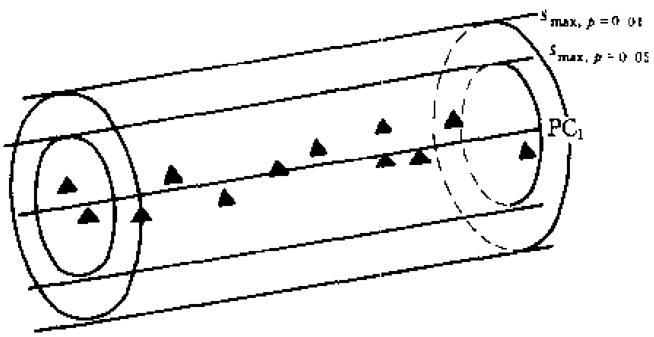


图 7-13 由一个主成分建立的类模型置信水平取 0.05 或 0.01

由于基于主成分分析的投影判别法主要是依靠人眼来进行判别分析的，为在主成分特征投影图将某一类的类模型形象给出，故 Wold 建议可直接从得分矢量入手，直接找到某一类在得分矢量上的上限与下限值，并以此来定义主成分的类模型。某一类在得分矢量上的上限与下限值可由下述式 (7-91) 算得

$$t_{lower,a} = t_{min,a} - (1/2)s_{t,a} \tag{7-91a}$$

$$t_{lower,a} = t_{min,a} - (1/2)s_{t,a} \tag{7-91b}$$

其中

$$s_{t,a}^2 = t_a^t t_a / n$$

在此，下标 a 表示第 a 个主成分。图 7-14 给出了一个由一个主成分建立的类模型例子。

对于任意未知样本 x_i^t，都可给出它的主成分模型：

$$x_i^t = x_c^t + \sum_k t_{ik} v_k^t + e_i^t \tag{7-92}$$

然后，计算它的误差项 e_i^t，继算出它到所定义的主成分模型的距离：

$$s_t = e_t^{\mathrm{t}} e_t / (d-A) \tag{7-93}$$

将它平方值与 s_{max}^2 进行比较，即用前述 F 检验即可判别它是否属于所定义的主成分模型。从此可以看出，只要求得了该未知样本矢量的误差项，判别就很容易了。所以，判别的过程，主

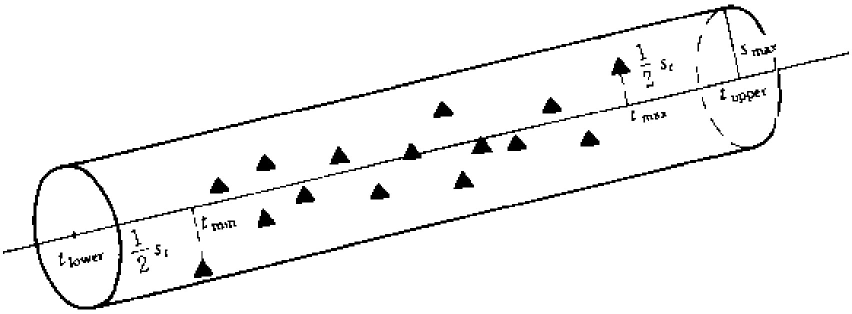


图 7-14　由极端得分值表述的由一个主成分构成的类模型

要是误差项的计算过程。未知样本误差项的计算可由下列步骤组成：

① 令 $e_t^{(0)} = x_t^{\mathrm{t}} - x_c^{\mathrm{t}}$

在此 x_c^{t} 为所求类的样本均值向量，为已知值。

② 对 $a=1, 2, \cdots, A$ 个主成分进行下述循环计算，以求得该样本在所求类的得分 t_{ta} $(a=1, 2, \cdots, A)$：

$$t_{ta} = e_t^{(a-1)} v_a$$

$$e_t^{(a)} = e_t^{(a-1)} - t_{ta} v_a$$

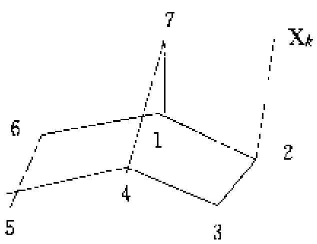
最终所得的 $e_t^{(A)}$ 即是未知样本误差项。继用下式算出它与所求主成分模型的距离：

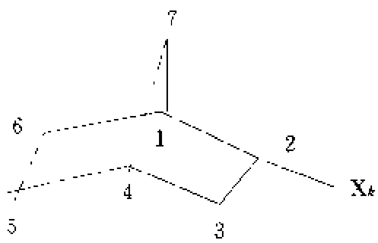
$$s_t = e_t^{\mathrm{t}} e_t / (d-A)$$

上述求未知样本误差项的过程始终都用到了主成分分析所得的荷载矢量的正交性质，即 $v_i^{\mathrm{t}} v_j = 0$ $(i \neq j)$。

下举一例说明 SIMCA 用于光谱数据解析的情况。欲以 ^{13}C 核磁共振（NMR）谱数据来判别降冰片烷的结构，这类化合物有内、外两种结构，能否利用 ^{13}C 核磁共振谱进行这种判别？训练集由 2 位取代的降冰片烷组成：

exo$(q=1)$　　endo$(q=2)$





k	X_k
1	CH_3
2	NH_3
3	OH
4	F
5	CN
6	COOH
7	$COOCH_3$
8	CH_2OH

k	X_k
9	CH_3
10	NH_3
11	OH
12	CN
13	COOH
14	$COOCH_3$
15	CH_2OH

用降冰片烷的 C_1～C_7 碳原子 ^{13}C 核磁共振谱化学位移 $\Delta\delta$ 作参量，$\Delta\delta$ 是该化合物与未取代降冰片烷（X_k＝H）的位移之差。本例中共有 15 个样本（$n=15$），7 个参量（$d=7$），2 个类（$q=2$）。先对原始数据进行均值中心化预处理，然后进行无监督的投影判别，可知两类均可用一个主成分模型表述（$A=1$），说明本例情况较为简单。根据特征参量对两类的判别能力，对特征参量进行了选择，发现只需保留 C_4、C_6、C_7 的化学位移 $\Delta\delta$，删去其他参量，对余下的特征参量重新计算得 t_{ia}、v_a 值。继根据 $s_i=e_i^t e_i/(d-A)$ 和 $s_c=s^t s/(n-A-1)$ 对诸样本和不同的类分别计算样本残余标准偏差 s_i 和类的均值标准偏差 s_c、F-检验证明，训练集的这些已知样本无不正常情况。用 SIMCA 法对未进入训练集的降冰片烷试样进行分类，能得到正确分类结果，甚至某些结构与降冰片烷近似的化合物也能给以分类，详见文献 [44]。

三、基于偏最小二乘的投影判别法

【基本思路】

如第六章所述，偏最小二乘方法本质上是一种基于特征变量的回归方法，在多元校正中得到了广泛地应用。偏最小二乘法与主成分分析的不同之点在于它不但对量测矩阵 $\boldsymbol{X}$ 进行正交分解，而且在分解 $\boldsymbol{X}$ 的同时对响应矩阵也进行正交分解，为一同时进行分解的特征变量回归法。如果将模式识别中已知类别的响应变量设为 0 或 1（对两类模型而言），或其他整数，如 1，2，3，…等（对多类模型而言），则偏最小二乘也可用于模式识别。特别是因偏最小二乘法也是一个特征变量法，所以，它也同时可用于特征投影以得到投影图，而将高维空间的样本点投影到低维空间进行人眼判别。由于 PLS 可同时对样本数据矩阵 $\boldsymbol{X}$ 和响应变量同时进行分解，并力图建立它们之间的回归关系，所得投影图似可得到更好的分类效果。

偏最小二乘法已经应用于很多不同的研究领域，在化学模式识别中也得到了相当广泛地应用。将偏最小二乘法用于有机合成、QSAR 和化学化工过程的优化是化学计量学研究的一个重要内容。

【几何意义及其应用讨论】

由于用于化学模式识别的偏最小二乘法的迭代过程也是通过由 H. Wold 提出的非线性迭代偏最小二乘算法而完成的。此在前一章已进行过详细讨论，所以在此从略，而只从几何意义方面进行一些必要的讨论。PLS 分解具有下述形式：

$$\boldsymbol{X}=\boldsymbol{USV}^t=\boldsymbol{U}^*\boldsymbol{S}^*\boldsymbol{V}^{t*}+\boldsymbol{E}_X=\boldsymbol{T}^*\boldsymbol{V}^{t*}+\boldsymbol{E}_X \qquad (7\text{-}94)$$

$$\boldsymbol{Y}=\boldsymbol{PGQ}^t=\boldsymbol{P}^*\boldsymbol{G}^*\boldsymbol{Q}^{t*}+\boldsymbol{E}_Y=\boldsymbol{R}^*\boldsymbol{Q}^{t*}+\boldsymbol{E}_Y \qquad (7\text{-}95)$$

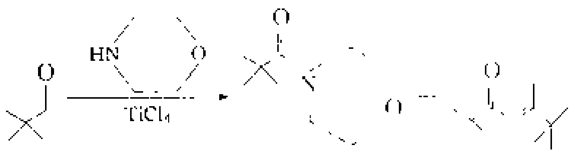
这里，由样本量测矩阵 $\boldsymbol{X}$ 分解得到的矩阵 $\boldsymbol{T}^*$ 以及由响应矩阵 $\boldsymbol{Y}$ 分解得到的矩阵 $\boldsymbol{R}^*$，代表了除去大部分噪声后的变量和响应的信息。经过这样的分解，我们可以得到两个分解的矩阵 $\boldsymbol{T}^*$ 和 $\boldsymbol{R}^*$，继将这两个分解的投影变量而不是原来的变量矩阵 $\boldsymbol{X}$ 和响应矩阵 $\boldsymbol{Y}$ 经回归联系起来。首先是变量矩阵的一个目标样本，即它的一个行矢量 $\boldsymbol{x}_i^t$，投影到 $\boldsymbol{v}_1^*$ 上，得到 t_{i1}，同理，响应矩阵的一个目标样本，即它的一个行矢量 $\boldsymbol{y}_i^t$，投影到 $\boldsymbol{q}_1^*$ 上，得到 r_{i1}，继对 t_{i1} 和 r_{i1} 进行回归，这样的关系在 PLS 中称为内相关。

偏最小二乘法的一个优点就是它可以在两个矩阵之间建立定量关系，这就使具有多个量测响应或活性矢量 $\boldsymbol{y}_j(j=1,2,\cdots,m)$ 的预测成为可能。设有化学量测、或结构信息、或试验变量矩阵，即 $\boldsymbol{X}$ 矩阵，同时还有多个类别响应或活性矢量 $\boldsymbol{y}_j$（$j=1$，2，…，m）所构成的响应矩阵 $\boldsymbol{Y}$，（$=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m]$），就可以用 PLS 来建立这两个矩阵之间的定量关系。比如，一个

化学反应可在一系列溶剂中进行，每个溶剂都可用几个特征变量表示，这就构成了变量矩阵 ***X***，用不同的溶剂，可以得到不同的主要产物的产率，其余副产物的产率等，这就构成了响应矩阵 ***Y***，借 PLS 就可建立它们之间的定量关系，从而，预测新的溶剂可能得到的结果。PLS 方法的第二个优点就是它可以运用不同的投影图来评价所得结果。对于大多数情况，对这些图的直观解释就可足够理解不同的变量与响应之间的关系了。

PLS 用于有机合成的例子——胺化反应的副产物的抑制在这个例子中，我们将讨论怎样应用 PLS 方法，在这个反应中找到调整哪一个变量就可以提高主要产物的产率而又能同时抑制副产物。

所研究的反应是用 3,3-甲基-2-丁酮来合成吗啉烯胺。本反应的副产物是 3,3-甲基-2-丁酮的自缩合：



试验设计矩阵与主要产物和副产物的产率所构成的矩阵就分别用作 PLS 方法的 ***X*** 矩阵和 ***Y*** 矩阵，它们都列于表 7-19。注意到，在 ***X*** 矩阵中实际还包括了 x 变量的平方项和交叉项。在做 PLS 之前，都对数据进行了预处理，即自标度化。

表 7-19　试验设计矩阵与响应矩阵

试验序号	变量			响应	
	x_1	x_2	x_3	y_1	y_2
1	−1	−1	−1	41.6	14.6
2	1	−1	−1	45.1	6.7
3	−1	1	−1	51.7	26.2
4	1	1	−1	61.7	17.7
5	1	−1	1	47.8	11.9
6	1	−1	1	57.1	7.5
7	1	1	1	63.0	26.1
8	−1	1	1	77.8	11.0
9	1.414	0	0	66.7	8.1
10	−1.414	0	0	49.5	22.2
11	0	1.414	0	70.4	18.9
12	0	−1.414	0	43.9	8.0
13	0	0	1.414	66.1	9.8
14	0	0	−1.414	52.4	17.3
15	0	0	0	56.5	13.8
16	0	0	0	60.0	12.3
17	0	0	0	58.6	12.6
18	0	0	0	57.2	13.6

① x_1——吗啉的用量；x_2——四氯化钛的用量；x_3——反应温度；y_1——烯胺的产率（%）；y_2——副产物的产率（%）。

经 PLS 分析，借交叉校验法，发现有两个主成分是显著的，它们代表了 ***Y*** 矩阵 96.2%的方差，第一个 PLS 主成分包括了 ***Y*** 矩阵 47.4%的方差，而第二个 PLS 主成分却代表了 48.8%的方差。

从数学上来解释 PLS 分析所得到的结果，***X*** 矩阵分解出的 V'^{*} 为：

$$V^{t*}=\begin{bmatrix}0.1814 & -0.9711 & -0.1317 & -0.0622 & 0.0315 & -0.2267 & 0.0326 & -0.0149 & 0.0057\\ 0.7863 & 0.1249 & 0.4986 & -0.1648 & -0.1241 & -0.0728 & 0.2199 & 0.0951 & 0.1111\end{bmatrix}$$

$$Q^{t*}=\begin{bmatrix}-0.6637 & -0.7480\\ 0.7600 & -0.6499\end{bmatrix}$$

这样、X 矩阵的得分矢量就可依据式（7-96）算出：

$$t_1=0.1814x_1-0.9711x_2-0.1317x_3-0.0622x_1^2+0.0315x_2^2 \\ -0.2267x_3^2+0.0326x_{12}-0.0149x_{13}+0.0057x_{23} \tag{7-96a}$$

$$t_2=0.7863x_1+0.1249x_2+0.4986x_3-0.1648x_1^2-0.1241x_2^2 \\ -0.0728x_3^2+0.2199x_{12}+0.0951x_{13}+0.1111x_{23} \tag{7-96b}$$

而

$$r_1=-0.6637y_1-0.7480y_2$$

$$r_2=0.7600y_1-0.6499y_2$$

两个得分矢量间的内部相关可表为

$$r_1=0.9751t_1+e$$

$$r_2=0.8802t_2+e$$

如果采用矩阵的表达方式，则有

$$R^*=BT^*+E \tag{7-97}$$

其中 B 为一对角矩阵：

$$B=\begin{bmatrix}0.9751 & 0\\ 0 & 0.8802\end{bmatrix}$$

注意到 PLS 的整个数学模型可表示如下：

$$X=USV^t=U^*S^*V^{t*}+E_X=T^*V^{t*}+E_X$$

$$Y=PGQ^t=P^*G^*Q^{t*}+E_Y=R^*Q^{t*}+E_Y$$

$$R^*=BT^*+E$$

得分矢量 r_i 与 t_i 的内部相关图示于图 7-15。

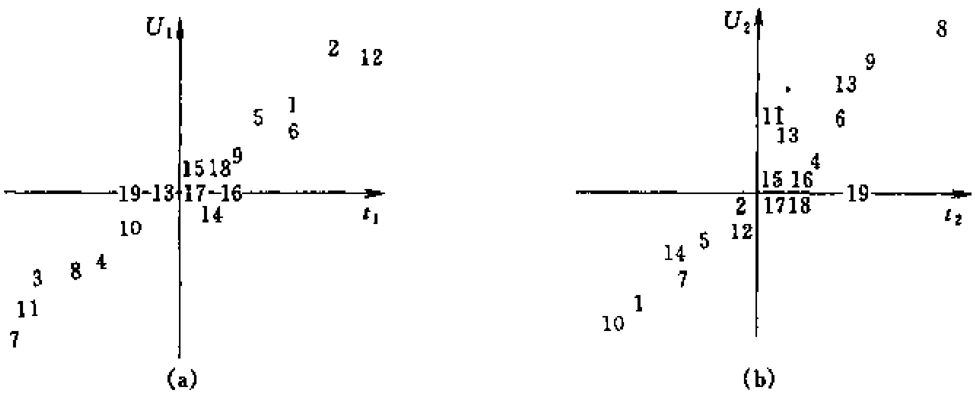


图 7-15　得分矢量 r_i 与 t_i 的内部相关图

(a) 第一个主成分；(b) 第二个主成分

从 Y 矩阵的荷载矩阵 Q^{t*}，可容易地计算出 Y 矩阵与得分矩阵的关系：

$$y_1=-0.6499r_1+0.7600r_2 \tag{7-98a}$$

$$y_2=-0.7480r_1-0.6637r_2 \tag{7-98b}$$

从式（7-98）可以看出，为了增加主要产物（烯胺）的产率，即 y_1，试验条件的调整应该是 r_1 朝负方向变化，而 r_2 朝正方向变化；而为了减少副产物的产率，即 y_2，试验条件的调整应该是 r_1 和 r_2 都朝正方向变化。这样，对于 r_1 来说，增加主要产物或减少副产物发生了矛盾。下面将可看到，这一矛盾是可以解决的。

由于 r_1 与 t_1 是平行的，所以可以通过调节 t_1 来代替 r_1。从 X 矩阵的荷载阵 Q' 可以看出，t_1 主要与 x_2 负相关（四氯化钛的用量）。正如 r_1，t_1 的减少将提高 y_1，这就意味着提高四氯化钛的用量将提高主要产物 y_1 的产率，但是，四氯化钛用量的提高也将冒提高副产物的风险，这是因为 t_1 的减少同样会提高 y_1；对于第二个 PLS 主成分 t_2，它主要与 x_1 和 x_3 正相关，这就说明如 r_2，t_2 增加，也即 x_1（吗啉的用量）和 x_3（反应温度）增加，将有利于主要产物的产率。以上结论与用响应面方法所得是一致的。

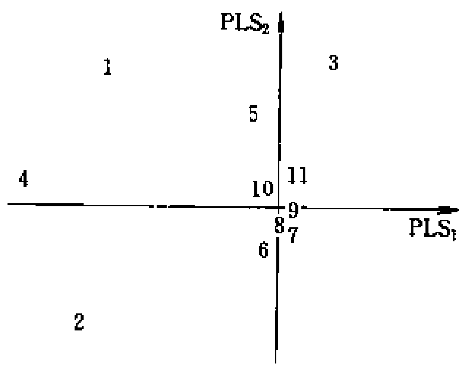


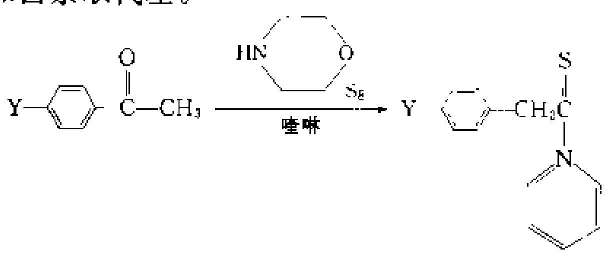
图 7-16 PLS 荷载投影图

1—y_1；2—y_2；3—x_1；4—x_2；5—x_3；6—x_1^2；7—x_2^2；8—x_3^2；9—x_{12}；10—x_{13}；11—x_{23}

另一方面，PLS 方法还可通过投影图来分析，而且它的分析比起上述的数值分析方法来得容易，来得直观。因为荷载投影图是反映变量之间的关系，所以只需将 q_1 对 q_2 及 v_1 对 v_2 同时作图（参见图 7-16），就可以得到类似结论。

从图 7-16 可以看出，1 和 2（y_1 和 y_2）的变化主要由 4 即 x_2 所决定（主要反映在 PLS_1 上），3 和 5（x_1 和 x_3）主要控制了 PLS_2 的变化，因为 1 即 y_1 处于正方向，故 3 和 5 即 x_1 和 x_3 的提高可提高 y_1 产率，而 2 即 y_2 处于负方向，故 3 和 5 即 x_1 和 x_3 的提高可降低 y_2 产率，这些结论与数值分析的结论是一致的。从这可以看出，PLS 投影图的分析比数值分析更容易更直观。

另一个例子是基于 Willgerodt-Kindler 反应，它同样说明了用 PLS 也可对离散变化进行逐步分析。早期的研究结果表明，对于强吸电子基团，如 NO_2、CN 等，而且，试验只局限于包括电子给予体、烷基和卤素取代基。



最初是选用了 5 个对位取代的乙酰苯（Y＝H、Me，Cl，MeO，Me_2N）组成了反应空间的取代基轴。最优产率的试验条件一般都是由响应面方法求出。试验结果示于表 7-20。为表征化合物，一般的取代参数等都收录于表 7-21。

首先，使用表 7-20 中所列的前 5 个化合物的数据作为 Y 矩阵，用所对应的表 7-21 中所列的 5 个化合物的结构参数数据作为 X 矩阵，来建立 PLS 的定量模型，对以氟为取代基的情况来作预测，所得结果列于表 7-20 的第六行，第七行列出了用响应面所得实验结果用以对照。然后，将氟为取代基的数据也包括进来，对溴取代的情况进行预测，所得结果列于第八行．第九行列出了相应的对照数据，从这些结果看来，PLS 方法的预测效果是很不错的。在这里，我们把溴取代的数据也包括进来，对甲硫基取代进行了预测，同时就按此条件做了试验，所得结果列子表 7-20 的第十行，从这一结果看来，用 PLS 预测的结果果然与实验结果吻合十分好

（预测 98%，实验 98%），说明 PLS 方法是一种很好的定量建模法。

表 7-20 Willgerodt-Kindler 反应的最优条件[①]

试验序号	取代基	试验最优条件			方法	产率	
		x_1	x_2	x_3		预测值	观察值
1	Me	9.6	8.4	133	RSM	96.0	96
2	MeO	9.3	8.9	130	RSM	98.2	100
3	H	7.5	10.3	123	RSM	90.8	94
4	Cl	9.7	9.9	119	RSM	95.0	100
5	Me_2N	8.8	8.3	122	RSM	89.0	89
6	F	7.8	11.0	112	PLS	89.0	
7	F	8.3	10.6	116	RSM	94.0	93
8	Br	10.4	9.3	123	PLS	98.4	—
9	Br	10.2	9.5	121	RSM	95.0	95
10	MeS	10.4	8.4	124	PLS	97.0	95

① x_1—硫/酮用量的比值（mol/mol）；x_2—吗啉/酮用量的比值（mol/mol）；x_3—反应温度。实验的产率是由气相色谱直接测出。

表 7-21 取代的结构参数[①]

取代基	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Me	−0.01	−0.17	−0.04	−0.13	−1.24	0	0.52	3.0	1.52	1.90	1.90	2.04	5.65	56
MeO	0.3	−0.27	0.26	−0.51	−0.55	−0.23	0.36	3.98	1.35	1.90	1.90	2.78	7.87	−0.02
H	0	0	0	0	0	0.32	0	2.06	1.00	1.00	1.00	1.00	1.03	0
Cl	0.47	0.23	0.41	−0.15	−0.97	−0.56	0.55	3.52	1.80	1.80	1.80	1.80	6.03	0.71
Me_2N	—	−0.83	0.10	−0.92	—	-	0.43	3.53	1.50	2.56	2.80	2.80	15.55	0.18
F	0.54	0.06	0.43	−0.34	−0.46	−0.14	0.27	2.65	1.35	1.35	1.35	1.35	0.92	0.14
Br	0.47	0.23	0.44	−0.17	−1.16	−0.84	0.65	3.83	1.95	1.95	1.95	1.95	8.88	0.86
MeS	0.3	0	0.2	−0.18	−1.07	−0.75	0.64	4.3	1.70	1.90	1.90	3.26	13.82	0.61

① 1—σ_1（Taft 诱导参数）；2—σ_p（对位取代基的 Hammett 参数）；3—F 和 4—R（Swain-Lupton 取代参数）；5—E_s 和 6—E_sc（Taft 立体参数）；7—v（van der Waals 半径）；8—L，9—B_1，10—B_2，11—B_3，12—B_4（Verloop 立体参数）；13—MR（分子折射率）；14—π（Hansch 亲脂参数）。

参 考 文 献

1 Diday E. Recent progress in distance and similarity measures in pattern recognition, Second international joint conference on pattern recognition, 1535～1539

2 陈民德，沈燮昌. 模式识别导论. 上海：上海科学技术出版社，1983

3 张尧庭，方开泰. 多元统计分析引论. 北京：科学出版社，1983

4 杨自强. 计算机应用与应用数学，1976；10：1～29

5 Kowalski B R, Bender C F. Anal Chem, 1972; 44: 1405

6 Coomans D, Massart D L. Anal Chim Acta, 1982; 136: 15

7 Coomans D, Massart D L. Anal Chim Acta, 1982; 138: 153

8 Coomans D, Massart D L. Anal Chim Acta, 1982; 138: 167

9 宋新华，陈茁，俞汝勤. 中国科学（B 辑），1993；23：245

10 Smits J R M, Melssen W J, Buydens L M C et al. Chemom and Intell Lab Sys, 1994; 23: 267

11 Zupan J, Gasteiger J. Anal Chim Acta, 1991; 248: 1～30

12 史忠福，神经计算. 北京：电子工业出版社，1993

13 Smits J R M, Melssen W J, Buydens L M C, et al. Chemom and Intell Lab Sys, 1994; 22: 165

14 张承福. 力学进展, 1994; 24: 186

15 焦李成. 神经网络计算. 西安: 西安电子科技大学出版社, 1993

16 马萨特 D L, 考夫曼 L 著. 刘思元译. 聚类分析法解析化学数据. 北京: 化学工业出版社, 1990

17 Rao C R, Cluster Analysis Applied to a study of race Mixture in Human Population, Classification and Clustering. Academic Press, 1977

18 Wishart W. Biometrics, 1969; 25: 165~170

19 Lance G N, Williams W T. Comp J. 1969; 12: 60~64

20 MacQueen J. Statistics and Probability, 1967; 1: 1

21 Ball G H, Hall D J. ISODATA, A novel method of data analysis and pattern classification, AD 699616, 1965

22 俞汝勤. 化学计量学导论. 长沙: 湖南教育出版社, 1991: 161~163

23 Babu G P, Murty M N. Pattern Recognition, 1994; 27: 321

24 Sun L X, Xu F, Liang Y Z et al. Chemom and Intell Lab Sys, 1994; 25: 51

25 Lucasius C B, Dane A D, Kateman G. Anal Chim Acta, 1993; 282: 647

26 Pop H F, Dumitrescu D, Sârbu C. Anal Chim Acta, 1995; 310: 269

27 Hibbert D B. Chemom and Intell Sys, 1993; 19: 277

28 Kalivas J. Simulated Annealing Applied to Chemistry, 1994

29 Zhang Q M, Yan K D, Tian S J et al. 中草药, 1991; 14: 15

30 Liu X D, Van Espen P, Adams F et al. Anal Chim Acta, 1987; 200: 421

31 金闻博, 戴亚. 烟草化学, 北京: 清华大学出版社, 1994: 50

32 Aruga R, Mirti P. Anal Chim Acta, 1993; 276: 197

33 Bezdek J C. Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum Press, 1983

34 Kim T, Bezdek J C, Hathaway R J. Pattern Recognition, 1988; 21: 651

35 Baldovin A, Wu W, Centner V et al. The Analyst, 1996; 121: 1603~1608

36 Wu W, Guo Q, Jouan-Rimbaud D et al. Chemom and Intell Lab Sys, 1999; 45: 39~53

37 Wold S, Sjostrom M. Chemom and Intell Lab Sys, 1998; 44: 3~14

38 Dief A S. Advanced Matrix Theory for Scientists and Engineers. Tunbridge Wells & London: Abacus Press, 1982, p. 93

39 Stewart G W. Introduction to Matrix Computations, New York: Academic Press, 1973, p. 340

40 Jellum E, Bjornson I, Nesbakken R, et al. J Chromatogr, 1981; 217: 231

41 Wold S, Albano C, Dunn W J, et al. Multivariate Data Analysis in Chemistry, in: Chemometrics: Mathmatics and Statistics in Chemistry, Ed Kowalski B R; NATO ASI Series C: Mathmatics and Physical Science, v. 138, D Reidel Pbb Co, Dordrecht, 1984: 17

42 Wold S, Sjostrom M. SIMCA: A Method for Analyzing Chemical Data in Terms of Similarity and Analogy, in: Chemometrics: Theory and Applications, Ed Kowalski B R ACS Ser, 1977: 52: 243

43 Wold S. Pattern Recognition, 1976; 8: 127

44 Yeung E S. Anal Chem, 1980; 52: 1120

第八章　计算机数字模拟法

本章介绍的计算机数字模拟法是研究化学反应、化学量测过程中的误差规律和进行其他化学过程的机理研究的一个很有效的辅助手段，亦属化学计量学研究的一个重要内容。本章将以两种不同的模拟方法为线索展开。首先，对于基于统计机理的Monte Carlo数字模拟法作出介绍。该类方法的主要特点是在于其概率特征，在化学动力学系统的模拟研究和误差分析的模拟研究中得到广泛应用，本章就主要以它们作为具体对象进行详细介绍。值得提出的是，对于很多别的分析过程，该方法亦可同样适用。所以，掌握该法的基本理论和基本思路比起具体计算机程序来就显得更重要了。本章将把着重点放在介绍该方法基本思路之上。对于基于微分方程数字解法的计算机模拟方法，本章在介绍目前在应用数学中常用的一些微分方程数字解法的基础上，主要对化学动力学及电化学过程的具体模拟作出详细介绍，值得提出的是，基于微分方程数字解法的计算机模拟的难点不在于其具体算法，而在于需模拟体系的微分方程本身及其初始条件和边界条件的确定。所以，本书的介绍重点是一些可通用的微分方程数字解法。

第一节　基于统计机理的Monte Carlo数字模拟法

二次世界大战期间，美国Los Alamos科学实验室的研究人员需要考察核反应的屏蔽问题，他们应用滚轮赌具试验构造理论模式，一步步逼近了问题的解。由于Los Alamos的研究项目当时是绝密的，采用著名赌城的名字Monte Carlo作为代码，从此，Monte Carlo模拟方法这一名词，成为随机模拟方法的代名词。

产生与组合具有特定统计特性的随机数是模拟动态随机过程的关键，这实际上就是随机变量本身的模拟[1]。随机变量服从一定的分布，可模拟产生服从所需分布的随机数。有时欲模拟体系中某一变量是一个确定性变量与一个随机变量的加合，亦可通过计算机来模拟。Monte Carlo模拟的基础是对任意已知分布的数学抽样，即在计算机上产生任意已知分布的随机变量的随机数。最容易得到的是（0，1）区间均匀分布的随机数 r。从均匀分布随机数，通过适当变换即可得到任意分布的随机数。然后介绍如何由均匀分布随机数产生任意分布的随机数。值得指出的是，在计算机上产生的随机数，是所谓“伪随机数”，即用数值方法产生的重复再现的（周期性的）“随机”数。取同样的起始值，得到的随机数系列总是全同的。

一、伪随机数的产生方法

（一）均匀分布的伪随机数的产生方法

对于计算机上产生的均匀分布的随机数系列，要求在指定区间均匀分布，即每个数有相同的出现概率，数字之间要实际上相互独立，并在指定的长度区间内次序不重复，但又具有一定周期性，而且，所用的算法应快速，占用内存少，下面将介绍两种最常用的方法。

乘同余法　乘同余法产生（0，1）区间均匀分布随机数序列的迭代计算公式为

$$x_i = a x_{i-1} \ (\mathrm{mod}\, m), \qquad r_i = x_i / m \tag{8-1}$$

在此，x_i 是伪随机数，i 是产生序列号，m 是一个用作模数的大的整数，a 是整常数。通过调

整 a、m 可控制产生的伪随机数系列的特性。上式的意思是：x_i 是 ax_{i-1}除以 m 后的余数，x_i/m 就是第 i 个在（0，1）区间均匀分布的随机数，如此反复迭代，可以产生一个随机数系列 r_1，r_2，…。

由于 x_i 是除数为 m 的除法中的余数，所以 $0\leqslant x_i\leqslant m$，即 $0\leqslant r_i\leqslant 1$。显然，不同的 x_i 值最多只能有 m 个，因此不同的 r_i 最多也只能有 m 个，即由乘同余法产生的系列 $\{r_i\}$ 具有周期性，其周期 $T\leqslant m$，产生 T 个数值后会发生循环。周期 T 的长短以及一个周期之内数目系列 $\{r_1, r_2, \cdots\}$ 随机性的好坏，与参数 a、m 和 x_0 的选取有关。

表 8-1　几个实际应用和检验过的参数值和周期

m	a	x_0	周期 T
2^{32}	5^{13}	1	$2^{30}\approx10^{9}$
2^{36}	5^{13}	1	$2^{34}\approx2\times10^{10}$
2^{42}	5^{17}	1	$2^{40}\approx10^{12}$

一般用半经验的方法来选择参数 a、m 和 x_0，使系列 $\{r_i\}$ 的周期比较长、随机性比较好。在计算机上实际产生出系列 $\{r_i\}$，然后用统计检验方法检验它的随机性，数值 r_i 在（0，1）区间上分布的均匀性及相互间的独立性。一些文献推荐下列类型的参数值：取 $x_0=1$ 或任意奇数，$m=2^k$、$a=5^{2q+1}$，k 和 q 都是正整数，则由迭代式（8-1）所产生的系列周期为 $T=2^{k-2}$。表 8-1 给出几个已经在计算机上实际应用和检验过的参数值和周期。

混合同余法　混合同余法的迭代计算公式为

$$x_i=ax_{i-1}+c\pmod m\qquad r_i=x_i/m \tag{8-2}$$

在此，x_i，i，m 和 a 的意义同前，c 为整常数，一般可取奇数。通过调整 a、c、m 可控制产生的伪随机数系列的特性。如取 $m=2^k$，$a=2^q+1$ 或 $a=4q-1$，x_0 取任意非负整数，可产生随机性好，且有最大周期 $T=2^k$ 的随机数系列 $\{x_i\}$[2,3]。

（二）正态分布的伪随机数的产生方法

方法 1　设 $r_1, r_2, \cdots, r_n$ 为 n 个相互独立的在（0，1）区间均匀分布的随机数，由于 $E(r_i)=1/2$，$D(r_i)=1/12$，根据李雅普诺夫中心极限定理推知，当 k 充分大时，

$$\eta_k=\sqrt{12/k}\left[\sum_{i=1}^{k}r_i-k/2\right] \tag{8-3}$$

的分布渐近于均值为零、方差为 1 的标准正态分布，记为 $N(0,1)$，故可把 η_k 近似地取为服从标准正态分布的随机数。通常 k 可取 10、11、12 等数，其中尤以取 12 为最简便。当 k 取 12 时

$$\eta=r_1+r_2+\cdots+r_{12}-6$$

再由线性变换

$$\omega=\alpha+\sigma\eta$$

即可得到均值为 α，均方差为 σ 的正态分布的随机数 ω。

方法 2　取两个独立且在(0,1)区间均匀分布的随机数 r_1，r_2 作变换

$$\eta_1=(-2\ln r_1)^{1/2}\cos(2\pi r_2) \tag{8-4a}$$

$$\eta_2=(-2\ln r_1)^{1/2}\sin(2\pi r_2) \tag{8-4b}$$

则 η_1, η_2 是两个独立的服从标准正态分布随机数。

再由线性变换

$$\omega_1=\alpha+\sigma\eta_1$$

$$\omega_2=\alpha+\sigma\eta_2$$

即可得到均值为 α，均方差为 σ 的正态分布的随机数 ω_1 和 ω_2。

由均匀分布的随机数还可以产生其他任意不同分布的随机数，如泊松分布、指数分布等，因它们在化学中不太常用，在此不再作详细介绍了，读者可参阅文献[4,5]。另外，在 MATLAB 程序语言中，都备有均匀分布和正态分布的随机函数，可以方便使用。

二、化学动力学系统的 Monte Carlo 模拟算法

基于化学动力学对待测物进行定性定量分析是分析化学的一个重要分支，所以，化学动力学系统的模拟在分析化学中占有重要地位。对化学动力学的研究，采用 Monte Carlo 模拟技术，可以避开解微分方程的困难，因而不需引入为解微分方程必须附加的限制条件或有关的稳态近似假设[6]。事实上，用 Monte Carlo 技术来研究和模拟化学动力学过程，就是将化学反应的有效碰撞过程看成是一个随机过程，在此随机过程中，化学反应的有效碰撞的几率将由反应的速率常数的大小和反应物的浓度来决定。本节将用一些微分方程易于描述的反应为例，以说明此法的基本原理[7]。

先考察最简单的单分子反应

$$A \xrightarrow{k_1} B$$

这里反应物分子 A 按一级反应分解为产物分子 B，放射化学中的裂变反应一般属于这种类型。为了用 Monte Carlo 方法模拟这一反应，可将计算机内存认作反应器，将分子 A（以数字 1 表示）存储于其中，具体而言，可设一个数组 $x(i)$，令其在指定的序号 i 内存储数字 1 代表分子 A。作为起始浓度，1 的个数代表了 A 分子的多少，即

$$[1]_0 \propto [A]_0$$

或

$$[1]_0 = a[A]_0 \tag{8-5}$$

式中，a 为比例常数。可另选用数字 0 代表不参与反应的溶剂等惰性分子，为了模拟反应容器的真实情况，应随机地将 1 与 0 存放在当作反应容器的计算机内存位置，即数组的不同随机序号中。但也可不必这样作，例如共 250 个分子 A（1），750 个分子 B（0），可分别放在 1～250 及 251～1000 位置上，而下面取用这些分子时，采用随机的办法，效果与随机存放是相同的。

先进行随机模拟。本例中，令程序产生 1～1000 的随机数。设出现的随机数为 31，即选择第 31 号位置上的元素，该元素是 1，反应发生，1 转化为 2，2 代表分子 B。它在反应前是不存在的，同时，记录一个反应时间。如下一个产生的随机数是 304，则不是数 1，而是代表惰性分子的 0，此时无化学反应发生。仍记录一个反应时间。如下一个随机数是另一个存有 1 的位置的序号数，则该处的 1 换成 2，又记录反应时间，余类推。设下一个随机数又出现 31，此时 31 处已是 2，同样无反应发生，因不考虑逆反应，同时记录反应时间。

然后对模拟过程进行分析。可以看出，分子 A 即数字 1 被选取——亦即发生反应的概率，是与置放它的内存空间及 A 的个数有关的。可按类似通常处理一级单分子反应的办法对这个模式进行数学处理：

$$-d[1]/dN_1 = \beta[1] \tag{8-6}$$

式(8-6)左边是数字 1 被 2 取代的速度，N_1 是单位时间内程序试图找出 1 的次数，方括号代表相应粒子的浓度，β 是只与反应“容器”大小有关的概率常数。显然，当 1 渐为 2 取代时，反应速度会下降。

将式（8-6）积分得

$$\ln\{[1]_0/[1]\}=\beta N_1 \qquad (8\text{-}7)$$

式(8-7)中的$[1]_0$是粒子 A 的初始“浓度”，$[1]$为其在时间 t 的浓度，考虑到式（8-5）及

$$[1]=a[A]$$

可以写出

$$\ln\{[A]_0/[A]\}=\beta N$$

试与一阶速度方程比较（速度常数 k_1）

$$\ln\{[A]_0/[A]\}=k_1t$$

可知

$$\beta N_1=k_1t$$

即一级反应速度常数与时间的乘积，等于相应概率常数乘以给定时间内程序试图寻找该粒子的次数。可见，一级反应的 Monte Carlo 模拟的时间单位与实际反应的时间之间的关系是很简明的。

上述模式易推广到较复杂的情况，例如序贯反应：

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

按常规方法处理

$$-d[A]/dt=k_1[A] \qquad (8\text{-}8a)$$

$$-d[B]/dt=k_1[A]-k_2[B] \qquad (8\text{-}8b)$$

$$-d[C]/dt=k_2[B] \qquad (8\text{-}8c)$$

当 $t=0$ 时，$[A]=[A]_0$，$[B]_0$ 及 $[C]_0$ 均为零，由此可得

$$[A]=[A]_0\exp(-k_1t)$$

$$[B]=\{k_1[A]_0/(k_2-k_1)\}[\exp(-k_1t)-\exp(-k_2t)]$$

$$[C]=[A]_0\{1+[1/(k_2-k_1)][k_2\exp(-k_1t)-k_1\exp(-k_2t)]\}$$

可见，此例的微分方程仍是可解的，虽然情况已较上一例复杂。现考察如何用 Monte Carlo 方法模拟，以数字 1 代表 A，数字 2 代表 B，仍以计算机的内存作反应“容器”，将数字存储于其中，显然，开始时仍只需存放数字 1。

$$-d[1]/dN_1=\beta_1[1]$$

$$-d[2]/dN_2=\beta_2[2]$$

由于 β_1、β_2 均只与反应“容器”大小有关，这里涉及的是同一容器，故二者相等，记为 β，将上述两式积分后，并采用与上例相同思路得

$$\beta N_1=k_1t$$

$$\beta N_2=k_2t$$

二式相除得

$$N_1/N_2=k_1/k_2$$

此式指出一个重要的事实，即用 Monte Carlo 方法模拟两个一级反应时，两个反应应按其速度常数值的比例进行模拟试验。即寻找 1 与 2 的次数之比应等于两个反应速度常数之比。对于较快的（速度常数大）的反应，应在单位时间内多模拟试验，这样模拟才能反映客观情况。例如，序贯反应

$$A \xrightarrow{k_1=4.0} B \xrightarrow{k_2=6.0} C$$

由 A 转化为 B 较 B 转化为 C 慢，这在模拟时应体现出来。进行这种模拟可用不同的方法，这里先介绍一种早期文献中介绍的一种方法，其思路有助于理解 Monte Carlo 方法的基本原理。找出 k_1 和 k_2 的最大公约数，它们的最大公约数为 2.0，就以 2.0 为计数单位进行模拟。由于每模拟 A→B 反应二次，应模拟 B→C 反应三次，可设置一个计数器 SN，先令 $SN=2.0$，然后检查 $SN \leqslant k_i$ 是否成立。本例中 $2.0 \leqslant 4.0$ 和 $2.0 \leqslant 6.0$ 均成立，将两个反应各模拟试验一次；然后，令 SN 增值，加 2.0 得 $SN=4.0$，再检查 $SN \leqslant k_i$ 是否成立，本例均仍成立，再将两个反应各模拟一次；再令 SN 增值，加 2.0 得 $SN=6.0$，此时，$SN \leqslant k_1$ 已不成立，停止模拟A→B反应，只模拟 B→C 反应一次，如此下去，每次检查 $SN \leqslant \mathrm{Max}\ (k_i)$ 是否成立，如此式成立，继续令 SN 增值；如 SN 已等于或大于 k_i 中数值最大者，则记录一次反应时间。用这样的方法，能实现在单位时间内，使 k_i 大的反应模拟次数较多，定量地反映出反应速率不同时的客观事实。用此 Monte Carlo 方法模拟序贯反应，所得结果与解微分方程法一致。用同样的原理模拟可逆反应、二级反应及复杂的反应，都能得到正确的结果。将该法用于离子电极跟踪的催化动力学反应[8]，取得了有价值的结果。

三、适用所有均相化学反应体系的 Monte Carlo 计算机模拟方法

以上介绍的模拟方法很直观，容易理解，但也存在一些弱点，如对于稍复杂的反应，用数组来存储反应粒子所需内存要大，另外，模拟复杂反应时，简单从数组中取活性粒子（反应物）或惰性粒子（溶剂或惰性化合物）来机械式地模拟化学反应所花时间也长。为此，Gillespie 根据将化学反应过程认作随机过程的假设，提出了颇具独到的新算法[9]，本书作者在此法的基础上，提出一种适用与所有均相化学反应体系的 Monte Carlo 计算机模拟方法[10]，现将其介绍如下。

（一）算法的基本原理

算法的基本原理是将计算机内存认作一反应器，其中存储了各物种的浓度，发生一次化学反应，对应的反应物或生成物就分别减少或增加 1 个计算机单位浓度，计算机记录这一变化同时记录下反应所需时间，为此需解决：① 对于存在 n 种物种与 m 条反应通路的反应体系，在下一时刻是哪一个反应发生？② 发生反应需多少时间？有无几乎同时发生的其他反应？如有，在模拟中如何体现？

1. 有效碰撞与分支反应

从反应速率的碰撞理论可知，一均相化学反应分支 i，即

$$A+B \xrightarrow{k} C+D$$

反应速率(dc_A/dt)完全由其反应分子的有效碰撞数 $a(i)$所确定，即

$$dc_A/dt = kc_Ac_B = a(i)$$

如体系中存在 m 个分支，则各分支反应的速率应与其有效碰撞数一致。设计算机每次反应的分子数一定，则在计算机模拟中下一时刻最大可能发生反应的应是有效碰撞数最大的分支反应。令计算机的模拟过程为一 Markov 随机过程，其游走由时刻 t 体系发生第 i 分支反应的概率 $P\ (i|t)$ 决定。显然，

$$P(i|t) = a(i)/a_0$$

其中

$$a_0 = \sum_{i=1}^{m} a(i) \quad (i=1,2,\cdots,m)$$

模拟中可产生一个(0,1)区间均匀分布的随机数 r，如有下列关系成立：

$$\sum_{i=0}^{s} P(i|t) < r \leqslant \sum_{i=1}^{s+1} P(i|t) = 1 \quad (s=0,1,2,\cdots,m-1)$$

式中，$P(i|t)$ 当 $i=0$ 时定义为零，则计算机进行第（$s+1$）分支反应。

2. 反应时间

（1）计算机“反应时间”与实际反应时间的换算　在模拟中，需假设一最低反应浓度，如所模拟的反应精度要求高，可将此值定小些，反之可定大些。

设进行计算机模拟的最低反应浓度为 $[M]$，即

$$[M]=\pm[x]=[x]_{t+\Delta t}-[x]$$

式中，$[x]$ 表示反应物时取负号，表示生成物时取正号，以保证 $[M]$ 恒为正值。确定了最低反应浓度即可算出一次反应所需时间 Δt

$$\Delta t=[M]/(kc_{A}c_{B})$$

如反应体系存在有 m 个反应分支，则计算机每进行一个分支反应所需时间为

$$\Delta t_i=[M]/a(i)$$

（2）偶合反应体系的“计算机时间”与实际反应时间的换算　对于稍复杂的化学反应体系，大都存在几个分支反应，而在计算机模拟中，每次只能进行一种反应。如何将此情况的计算机时间正确地换算成为实际体系进行反应所需时间是实现该模拟算法的关键。根据 Ostwood 反应独立共存原理[11]，即任一反应的速率与其他反应的存在无关，存在多分支反应时，每一瞬间应有 m 个分支反应几乎同时发生，可分两种情况进行讨论：

① 设在反应体系中同时存在 m 个分支反应，经 n 次计算机进行的化学反应后，m 个分支各自所需时间为

$$t_i=n_i\Delta t_i=n_i[M]/a(i) \quad (i=1,2,\cdots,m)$$

式中，n_i 表示在总数为 n 次计算机模拟反应中第 i 分支反应所发生反应的次数，有 $\sum n_i=n$；$a(1)=k_1C_{xj}C_{xk}$；$a(2)=k_2C_{xl}C_{xm}$；…，如存在偶合，则有下标($j,k,l,m,\cdots$)中任意两个相等的情况。

因 $n_i \propto P(i|t)$，故有式(8-9)成立

$$n_1/a(1) \approx n_2/a(2) \approx \cdots \approx n_{m-1}/a(m-1) \approx n_m/a(m) \tag{8-9}$$

实际模拟时，取 $n_i[M]/a(i)$($i=1,2,\cdots,m$)的均值作为进行了 n 次计算机模拟之实际反应所需时间。

② 如在反应体系中还存在连串反应或自催化反应，且它们之间的速率常数相当悬殊，则在计算平均时间时需删去一些分支反应的时间，并认为此时体系的反应时间应完全由其反应速度的决定步骤来决定。如有反应：

$$\mathrm{A} \xrightarrow{k_1} \mathrm{B} \xrightarrow{k_2} \mathrm{C} \qquad k_2 \gg k_1$$

$$a+b \xrightarrow{k'_1} c;\ c+a \xrightarrow{k'_2} d \qquad k'_2 \gg k'_1$$

上述两个反应的反应速度都是由第一个分支反应决定，即此时 $n_1/a(1)$ 与 $n_2/a(2)$ 不可能近似相等。因在复杂反应体系中上述情况普遍存在，故计算反应体系的实际反应时间时，不

能对所有的分支反应时间同等对待，而应先剔除某些分支反应时间很小的值，在对剩下的分支反应时间取均值。

（二）模拟算法

模拟算法由下述步骤组成：

第一步，输入反应体系的所有速率常数及各物种的初始浓度。赋 $t_{总}$ 初值为零。

第二步，赋各个分支反应时间 $t_i(i=1,2,\cdots,m)$初值为零。

第三步，赋 n_1 初值为零。此步的意义是使每次输出数据时，至少进行了 500 次，以保证必要的计算精度。

第四步，计算游走概率 $P_i(i=1,2,\cdots,m)$，产生(0,1)区间均匀分布的伪随机数，根据随机数来决定下一步反应的进行。

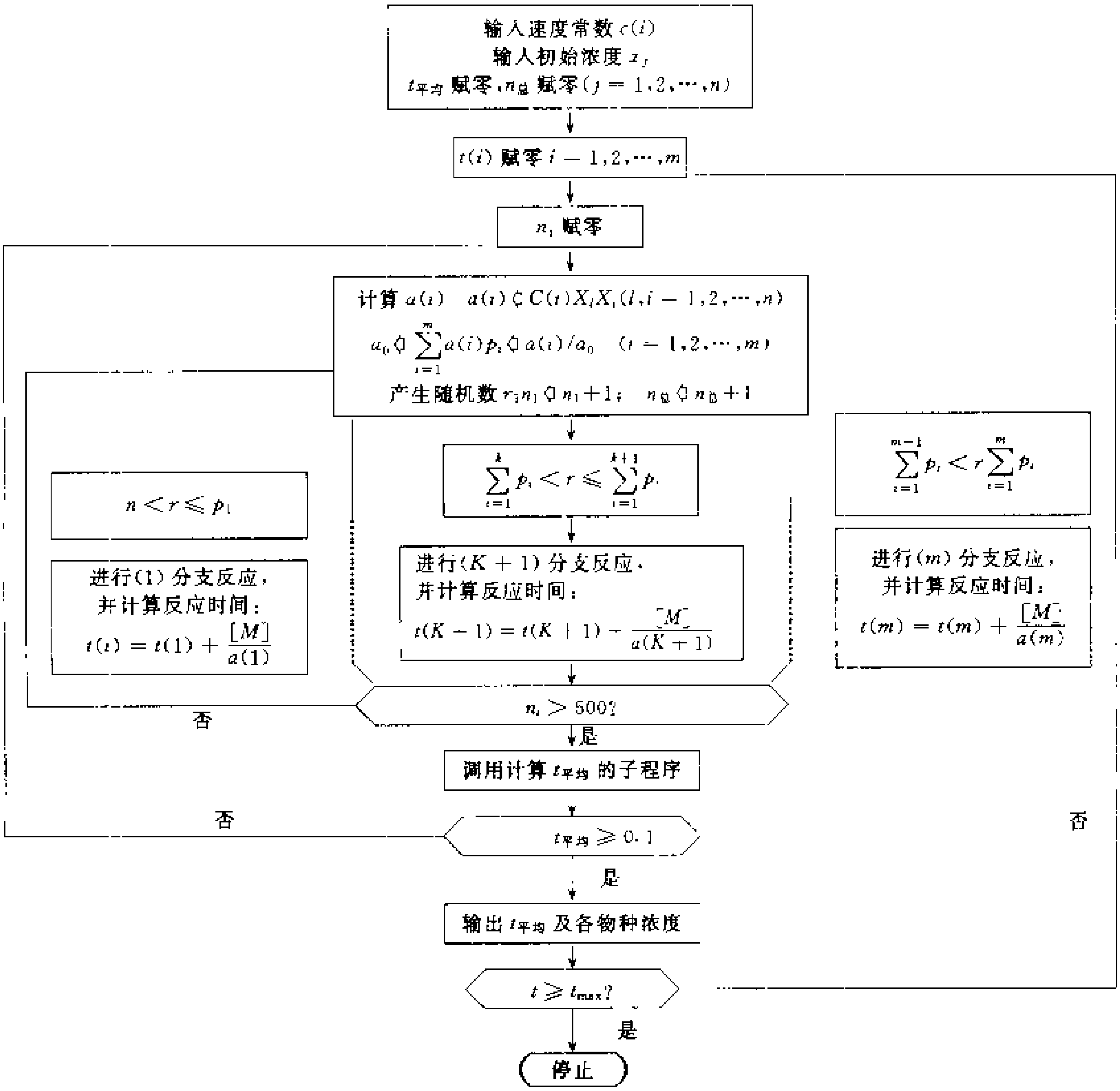


图 8-1　化学动力学 Monte Carlo 模拟程序框图

第五步，进行第 i 个分支反应，并计算该反应进行所需时间。如果 n_1 大于 500，用子程序剔除非决定步骤的分支反应时间，然后计算 $t_{平均}$，如此时 $t_{平均}>0.1\text{s}$，则此时输出各物种浓度值，否则返回第三步继续运算，直至 $t_{平均}>0.1\text{s}$ 时输出数据，输出数据后返回第四步，如此

往返。

第六步，比较 $t_{总}$ 是否达到所要求的反应时间，如未达到，返回第二步，直至 $t_{总}$ 达到所要求的反应时间 t_{max}时则停止。

计算机模拟程序框图示于图 8-1。

用此法对著名的 Belosov-Zhabotinsky 化学振荡反应的 Oregonator 模型[12]进行了模拟，获得满意结果[10]。值得指出的是，仅用模拟方法不能对有关反应机理作出最后肯定的判别，模拟方法能发现有的假设的机理与实验结果不符，从而否定这种假设。但在模拟预测的结果与实验一致时，应进一步作深入的研究，例如，设法找出有关中间产物的直接实验证据。

四、误差分析的 Monte Carlo 模拟算法

任何化学量测技术与相应数据处理步骤均涉及误差问题，误差的大小和不同分布对分析结果将产生重要影响。各种仪器的误差的不同特性和大小原则上可通过实验的方法，即通过反复分析已知试样或标样来获得，可惜此法耗财费时，且不易分别查明各个环节与因素是怎样通过误差来影响分析结果的，此时，如采用计算机模拟方法就显得比较灵活且方便，它不但可以模拟各种不同分布的误差将怎样影响结果，而且还可考察出测量过程中任意步骤对结果影响的大小。采用前节介绍的随机数发生技术，为计算机模拟误差分析提供了有力手段。下面以离子选择性电极两次加入法[13]为例来加以具体说明。

不少评论家认为这一方法误差太大。此法的要点是：先用选定的离子电极测试含待测离子 x（浓度 c_x）的试液

$$E_0=E_s+s\lg(c_x) \tag{8-10}$$

作标准加入，使 c_x 增加 Δc_1 及 Δc_2，每次加入后测试电位值：

$$E_1=E_s+s\lg(c_x+\Delta c_1) \tag{8-11}$$

$$E_2=E_s+s\lg(c_x+\Delta c_1+\Delta c_2) \tag{8-12}$$

$E_i(i=1,2)$为测得的电位值，E_s 是电极标准电位，s 是电极的斜率。合并（8～10）～（8-12）三式，得

$$(E_2-E_1)\lg[(c_x+\Delta c_1)/c_x]-(E_1-E_0)\lg[(c_x+\Delta c_1+\Delta c_2)/c_x]=0 \tag{8-13}$$

对（8-13）式难于求得的显式解，现用 Monte Carlo 模拟法对此式进行考察。作标准加入测定时，标准液浓度的误差、容器误差等实验误差及量测 E_0、E_1、E_2 的测试误差，均会影响分析结果。现分成几步一一模拟。

① 首先，产生正态分布的随机数（$\Delta c_1)_N$ 及（$\Delta c_2)_N$，令其均值相应为 Δc_1、Δc_2，相对标准差为 RSD_Δ，此相对标准差表征了 Δc_i 的误差，主要由容器误差造成。

② 用设定的 c_x 和随机产生的正态分布的随机数（$\Delta c_1)_N$ 及（$\Delta c_2)_N$，借前述式子计算 E_0、E_1、E_2，显然，E_0 是对应于给定的 c_x 值的理论电位值，而 E_1、E_2 中已包含 Δc_1 及 Δc_2 的误差。由此计算得到的 E_0、E_1、E_2 供下一步模拟使用。

③ 产生正态分布的随机数$(E_0)_N$、$(E_1)_N$ 及$(E_2)_N$，令其均值相应为前一步计算的 E_0、E_1、E_2，标准差为 σ_E。这里，为了简化模拟，设所有电位量测误差 σ_E 相同，但也可在模拟时给定不同的 σ_E，例如量测 E_0 时，浓度较量测 E_1、E_2 时低，可能 E_0 的量测相对误差较大。

注意②～③步模拟了 Δc_1 的误差传递到 E_1，E_1 又有本身误差；Δc_2 的误差传递到 E_2，E_2 又有其本身误差，下一步将传递到计算上去。

④ 将$(E_0)_N$、$(E_1)_N$ 及$(E_2)_N$ 代入式(8-13)中作为 E_0、E_1、E_2 值，用 Δc_1 及 Δc_2 的标称值

代入（8-13）式中，用Newton-Raphson法求解（8-13）式，这一步模拟得到的是$(c_x)_N$。这里Δc_1及Δc_2用的是实验中所取的标称值，而不是加入了误差的模拟值，这是因为在实际的二次标准加入法计算中就是这样作的。

⑤ 将①～④步重复多次，例如500次，计算$(c_x)_N$的相对标准差RSD。所得的500个数据可给出$(E_0)_N$、$(E_1)_N$、$(E_2)_N$及$(c_x)_N$的分布图。$(E_0)_N$、$(E_1)_N$、$(E_2)_N$自然是服从正态分布的，模拟结果表明，$(c_x)_N$亦确是服从正态分布的。如取c_1、c_2的相对标准差$RSD_{\Delta c}$为0.7%，电位量测误差0.1mV，取未知浓度为$c_x=10.0$，$\Delta c_1=10.0$，$c_2=40.0$，得$RSD=3.3\%$。取较大的$\Delta c_1/c_x$与$\Delta c_2/\Delta c_1$值可增加精密度，使c_x的相对标准差RSD最低可达1.5%。但用这一方法有一定限度，因Δc_1、Δc_2取得过大，将使二次标准加入法的一些基本前提不再成立，这些前提是加入Δc_1、Δc_2后体系的离子强度等条件无显著变化，离子活度系数及电极的参量E_s、S无变化。

Monte Carlo方法不仅可用于化学量测误差的模拟，它还可用于许多化学计量学算法本身特性的考察，目前很多新方法的发展大都需用Monte Carlo法先进行模拟研究的原因就在于此。用Monte Carlo模拟方法研究因子分析中主成分数的一些判别标准的特性也是一例[14]。有关Monte Carlo方法及其应用的详细介绍，可参阅文献［15］，该书对此给出了很好的阐述。

第二节　基于微分方程数字解法的计算机模拟方法

一、化学动力学系统的微分方程数字解法

一个化学体系的动力学特征一般都可由一组常微分方程来表征，所以，如果需深入了解这个化学体系的动态过程，一般是对该常微分方程组直接求解。然而，能精确求解的微分方程是寥寥无几的，所以我们常常希望求近似的数值解。本节将介绍的就是这样一些常用的数值求解常微分方程的方法。这些方法的基本思路是将时间变量离散化，即产生一离散点序列t_0，t_1，t_2，t_3，…，以及变量区间$h_n=t_{n+1}-t_n$。然后，在每一点t_n处，用先前求得的值计算，并用该计算值作为该时间点解的近似值。这样的方法在微分方程数值解方法中称为逐步法（也称差分法或离散变量法）。差分法提供了一个利用k个先前得到的数值y_n，y_{n-1}，y_{n-2}，…，y_{n-k+1}来计算y_{n+1}的法则，称为k步法。如果$k=1$，则是单步法，如果$k>1$，则是多步法。

欧拉法是最简单的单步法。在欧拉法中，y_{n+1}的值由前一点的值y_n直线外推求得。考虑单个方程

$$\mathrm{d}y/\mathrm{d}t=y'=f(y,t)$$

假设给定初值$y(t_0)=y_0$，这样，$y(t_1)$的近似值y_1就可以利用$y(t_0)$处的台劳级数展开式的前两项求得，即

$$y(t_1)\approx y_1=y_0+h_0f(y_0,t_0)$$

然后，令$t_2=t_1+h_1$，并计算

$$y(t_2)\approx y_2=y_1+h_1f(y_1,t_1)$$

按此法则继续下去。于是，一般形式有

$$y_{n+1}=y_n+h_nf(y_n,t_n)$$

然而，尽管求常微分方程的数值解的方法很多，但它们通常都能归入以下介绍的方法中的一种。下面，将对常用的微分方程数值解法以其不同特性为线索加以简要介绍。

（一）台劳级数法

如果解 $y(t)$ 是光滑的，那么它就具有以下的台劳级数展开式：

$$y(t+h)=y(t)+h\mathrm{d}y/\mathrm{d}t+(h^2/2!)(\mathrm{d}y^2/\mathrm{d}t^2)+\cdots \tag{8-14}$$

前述的欧拉法可以看做为一种取台劳级数前两项的近似方法。如果能求得 y 的高阶导数，即

$$y_{n+1}=y_n+hy_n'+(h^2/2!)y_n''+(h_3/3!)y_n'''+\cdots+(h^p/p!)y_n^{(p)} \tag{8-15}$$

式中，y_n'、y_n'' 及 $y_n^{(p)}$ 分别表示解函数 $y(t)$ 在第 n 个离散点时的一阶、二阶及 p 阶导数。那么，利用上式就可以得到 p 阶方法。局部离散误差可以根据舍去的第一项来估计，步长也可由此选定。对于光滑可微解函数，台劳级数方法一般说来是很有效的。

（二）龙格-库塔法

龙格-库塔法（Runge-Kutta Method）为近似的台劳级数法，它只需知道一阶导数而勿需明确定义或计算其他阶导数。近似值由若干次函数求值得到。经典的四阶龙格-库塔法由（8-16）式给出：

$$y_{n+1}=y_n+1/6[k_0+2k_1+2k_2+k_3] \tag{8-16}$$

式中

$$k_0=hf(y_n,t_n)$$
$$k_1=hf(y_n+1/2k_0,t_n+1/2h)$$
$$k_2=hf(y_n+1/2k_1,t_n+1/2h)$$
$$k_3=hf(y_n+k_2,t_n+h)$$

注意，对于每一步必须求函数 $f(y,t)$ 的值 4 次。

当 $h\to 0$ 时，经典的四阶龙格-库塔法通过阶为 h^4 的各项渐近地与台劳级数法相吻合。然而，没有任何对局部离散误差的估计来确定步长。龙格-库塔法有几个优点。它编程容易且对很大一类问题数值计算稳定。因为该法只需要一个解 y_n 就能算出 y_{n+1}，所以这种方法能自动起步，步长 h_n 在计算过程的每一步都可改变。对于龙格-库塔法曾有过不少改进，以提高其精度，在此介绍一个五阶方法[16]。

五阶龙格-库塔法的计算公式：

$$y_{n+1}=y_n+h/192[23k_1+125k_3-81k_5+125k_6] \tag{8-17}$$

而

$$k_1=f(y_n,t_n)$$
$$k_2=f(y_n+h/3k_1,t_n+h/3)$$
$$k_3=f(y_n+h/25[4k_1+6k_2],t_n+2h/5)$$
$$k_4=f(y_n+h/4[k_1-12k_2+15k_3],t_n+h)$$
$$k_5=f(y_n+h/81[6k_1+90k_2-50k_3+8k_4],t_n+2h/3)$$
$$k_6=f(y_n+h/75[6k_1+36k_2+10k_3+8k_4],t_n+4h/5)$$

（三）多步法

上述所讨论的方法都是仅通过取决于 t_n，y_n 和步长 h_n 的函数来计算 y_{n+1} 值的。显然，若利用先前求得的点的信息，即 y_{n-1}，y_{n-2}，y_{n-3}，…和 f_{n-1}，f_{n-2}，f_{n-3}，…，应可以得到更好的精度（在此，$f(y,t)=\mathrm{d}y/\mathrm{d}t=y'$）。基于这种想法的多步法是非常有效的。相对于单步法来说，当要求精度高时，它常常只需要较小的工作量，而且很容易获得截尾误差的估计，如果程序设计的恰当、合理，那么多步法能够不改变 h 值而在任意点有效地输出结果。这一方法的阶

可以自动选择和动态改变，从而提供了一种能处理范围很广的问题的方法。刚性方程也可用某些多步法来处理，多步法还能对方程自动地进行刚性与非刚性分类。然而，得到这些优点是以程序的复杂性和在某些情况下数值上不稳定为代价的。

线性多步法可以认为是下述公式(8-18)

$$y_{n+1}=\sum_{i=1}^{k}\alpha_i y_{n-1-i}+\sum_{i=0}^{k}\beta_i f_{n-i} \tag{8-18}$$

的特殊情况。这里，k 是一个确定的整数，α_i 或 β_i 不为 0，该式(8-18)定义了一般线性 k 步法。这一方法之所以称为线性的，是因为上面的多步法公式是 f_i 的线性函数，而 f 本身则可能是也可能不是它的自变量的线性函数。

在这个方法"启动"以后，每一步都根据已知值 $y_{n-1}, y_{n-2}, \cdots, y_{n-k-1}$ 和 $f_{n-1}, f_{n-2}, f_{n-3}, \cdots, f_{n-k+1}$ 来计算 y_{n+1} 的值。如果 $\beta_0=0$，则这种方法是一种显式方法，计算是直接的。若 $\beta_0\neq 0$，则这种方法是一种隐式方法，因为它需要 f_{n+1} 的值方能得 y_{n+1}。如隐式具有更多的所要求的特性，因而应用隐式方法时存在较大的困难也是理所当然的。

通常，在求解的每一步都使用两种多步法。首先使用显式法进行预报，然后再运用一次或多次隐式法加以校正——因此被称为预报-校正法。预报-校正法的一个很好的例子是四阶 Adams 法[18]。

预报：$$y_{n+1}=y_n+h/24[55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3}] \tag{8-19}$$

校正：$$y_{n+1}=y_n+h/24[9f_{n+1}+19f_n-5f_{n-1}-f_{n-2}] \tag{8-20}$$

式(8-19)，式(8-20)两个公式都是四阶的。由于多步法需运用前几个计算值才能进行计算，所以，在实际编程时，大都采用龙格-库塔法起步，计算三步后，再采用上述两个公式进行多步法计算。

另一个多步法的例子是定步长哈明（Hamming）方法[18]，其计算公式如下：

$$y_{n+1}=c_{n+1}+9/121[p_{n+1}+c_{n+1}] \tag{8-21}$$

式中

$$p_{n+1}=y_{n-3}+8h/3[f_n+f_{n-2}-f_{n-1}]$$
$$c_{n+1}=1/8\{9y_n-y_{n-2}+3h[f(r_{k+1})+2f_n-f_{n-1}]\}$$
$$r_{k+1}=p_{n+1}+112/121[p_n-c_n]$$

同 Adams 法，也可采用龙格-库塔法起步，计算三步后，再采用上述公式（8-21）进行多步法计算。

（四）刚性方程的吉尔法（Gear Mehtod for Stiff Equation）

刚性方程是一类很特殊的微分方程，因其很难用一般的微分方程的数值解法来解，故一直是微分方程解法研究的一个重点问题，有效的方法还推 Gear 法[19]。对于方程组

$$\boldsymbol{y}'=\mathrm{d}\boldsymbol{y}/\mathrm{d}t=\boldsymbol{f}(t,\boldsymbol{y})=\boldsymbol{f} \tag{8-22}$$

式中

$$\boldsymbol{y}=\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$f=\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$

则微分方程组的解的稳定性与雅可比矩阵 $\partial f/\partial \boldsymbol{y}$ 的特征值有关。例如，考察

$$f_1=\mathrm{d}y_1/\mathrm{d}t=998y_1+1998y_2 \tag{8-23a}$$

$$f_2=\mathrm{d}y_2/\mathrm{d}t=-999y_1-1999y_2 \tag{8-23b}$$

则雅可比矩阵 $\partial f/\partial \boldsymbol{y}$ 为

$$\partial \boldsymbol{f}/\partial \boldsymbol{y}=\begin{bmatrix} \partial f_1/\partial y_1 & \partial f_1/\partial y_2 \\ \partial f_2/\partial y_1 & \partial f_2/\partial y_2 \end{bmatrix}=\begin{bmatrix} 998 & 1998 \\ -999 & -1999 \end{bmatrix} \tag{8-24}$$

它的特征值是 -1 和 -1000。如果 $y_1(0)=y_2(0)=1$，则此微分方程组(8-23)的解析解为

$$y_1=4\mathrm{e}^{-t}-3\mathrm{e}^{-1000t}$$

$$y_2=-2\mathrm{e}^{-t}+3\mathrm{e}^{-1000t}$$

即经过一段很短的时间后，解就非常接近于

$$y_1=4\mathrm{e}^{-t}$$

$$y_2=-2\mathrm{e}^{-t}$$

如 t 大于 0.01 时，$y_1=4\mathrm{e}^{-0.01}$，$y_2=-2\mathrm{e}^{-0.01}$，两个解都接近于零了。可是如果采用欧拉法解微分方程组(8-23)，则其离散解可写成

$$y_{1,n+1}=y_{1,n}+h(998y_{1,n}+1998y_{2,n})$$

$$y_{2,n+1}=y_{2,n}+h(-999y_{1,n}-1999y_{2,n})$$

如在此仍取 $y_1(0)=y_2(0)=1$，$h=0.01$，则有

$$y_{1,1}=1+0.01(998+1998)=30.96$$

$$y_{2,1}=1+0.01(-999-1999)=-28.98$$

再继续积分几步，其结果将变得更加糟糕。此时，可通过较小的 h 值而使问题得到部分改善，但最终将由于舍入和截断误差的充分积累而得到另一种不稳的结果。因此，前述的大多数标准方法都不太适合解刚性方程。

可是，在化学动力学的研究中，很多微分方程还属于刚性方程的范畴，如一些含有自催化过程的复杂反应、化学振荡反应等，其微分方程组大都属于刚性方程之列[20]，所以，关于刚性方程的数值解法还有必要在此介绍。

考虑如式（8-22）所示的常微分方程组的初值问题，记

$$\boldsymbol{Z}_k=[\boldsymbol{y}_k, h\boldsymbol{y}_k', \cdots, h^q/(q!)\boldsymbol{y}_k^{(q)}]^{\mathrm{T}}$$

可知 $\boldsymbol{Z}_k$ 是一个 $(q+1)\times n$ 的矩阵。

对于 Adams 预估-校正方法的计算公式为

$$\boldsymbol{Z}_{k+1}^{(0)}=\boldsymbol{P}\boldsymbol{Z}_k \tag{8-25a}$$

$$\boldsymbol{Z}_{k+1}^{(m+1)}=\boldsymbol{P}\boldsymbol{Z}_{k+1}^{(m)}-\boldsymbol{d}[h\boldsymbol{f}(t_k,\boldsymbol{y}_k^{(m-1)})-h\boldsymbol{f}(t_k,\boldsymbol{y}_k^{(m)})]^{\mathrm{T}} \quad (m=1,2,\cdots,M) \tag{8-25b}$$

$$\boldsymbol{Z}_{k+1}=\boldsymbol{P}\boldsymbol{Z}_{k+1}^{(M)} \tag{8-25c}$$

在此，$\boldsymbol{P}$ 是 $(q+1)\times(q+1)$ 阶的帕斯卡耳(Pascal)三角矩阵，其元素为

$$p_{ij}=C_j^i \quad (q\geqslant j\geqslant i\geqslant 0)$$

M 为迭代次数，一般可取 3。向量 $\boldsymbol{d}$ 为

$$d=[d_0, d_1, \cdots, d_q]^{\mathrm{T}}$$

是一个预置值的常数向量，因方法不同而预置值不同。

由于$[hf(t_k, y_k^{(m-1)})-hf(t_k, y_k^{(m)})]^{\mathrm{T}}$为一(1×$n$)阶的行向量，所以，$d[hf(t_k, y_k^{(m-1)})-hf(t_k, y_k^{(m)})]^{\mathrm{T}}$就是一个$(q+1)\times n$的矩阵。

对于刚性方法的计算公式为

$$Z_{k+1}^{(0)}=PZ_k \tag{8-26a}$$

$$Z_{k+1}^{(m+1)}=PZ_{k+1}^{(m)}-d\{G[hf(t_k, y_k^{(m-1)})-hf(t_k, y_k^{(m)})]\}^{\mathrm{T}} \quad (m=1,2,3,\cdots,M) \tag{8-26b}$$

$$Z_{k+1}=PZ_{k+1}^{(M)} \tag{8-26c}$$

但是，$G=[d_1 I-d_0(\partial f/\partial y)]^{-1}$，其中$I$为$(n\times n)$阶的单位矩阵，$(\partial f/\partial y)$即雅可比矩阵。有关Gear方法的Fortran子程序可参阅文献[5]。

二、电化学过程的数字模拟

电化学包括电分析化学中的许多复杂电极过程是数字模拟的研究对象，用模拟方法研究电化学传输过程已有较长的历史。电化学传输过程中涉及的最基本的关系式是Fick第二定律表述的扩散方程：

$$\partial c/\partial t=D\partial^2 c/\partial x^2 \tag{8-27}$$

式中，c为浓度，x为位移。

(8-27)式是Fourier研究的热传导方程的一个应用方面，是抛物型的二阶偏微分方程。由于电化学过程中还涉及对流、均相的或非均相的化学反应等引起的浓度变化的复杂因素，用解析法来求解得到的偏微分方程极为困难，所以，数字模拟是获取答案的重要方法。解这类描述电化学过程的偏微分方程一个基本的方法是有限差分法。Feldberg等最早研究用计算机进行电化学过程的数字模拟[21]，他撰写的有关"数字模拟——解电化学扩散动力学问题的通用解法"论文[22]，是这一领域的经典著作。本节将简介Feldberg提出的"箱法"及Britz发展的"点法"，而且，本节的叙述只介绍有关方法的基本思路，对于一些细节，可参阅Britz撰写的有关电化学中数字模拟的专著[23]。

对电化学过程考察式(8-27)，首先涉及的问题是以电极表面为起点，计算溶液中距电极表面不同距离x处在给定时间t时有关物质的浓度。由于用解析方法求解有关偏微分方程的困难，试模拟物质由电极表面到溶液中的实际扩散等过程。这里将以x坐标表述空间，离散为长度h的小区间，时间坐标t则离散为长度δt的小区间，即式(8-27)中的x与t相应以h及δt的倍数表述：

$$x_i=ih$$

$$t_i=i\delta t$$

如绘出t-x图，图上各代表浓度c值的点构成一个网络（图8-2)。

考察图8-2中标出的点，所谓数字模拟问题就是找出每一时间间隔δt下沿x轴的浓度系列。Feldberg[22]研究的方法的基本思路，是设想从一维方向观察一很细的溶液体，其截面积为A，以等长h分割成为许多"箱"单位，如图8-3所示。试考察3个相邻的"箱"单元，$i-1$，i，$i+1$。在时间间隔δt，有溶质由$i-1$单元流入i，同时也有溶质由i流出，进入$i+1$单元，依Fick第一定律，沿观测方向x溶质扩散通过截面积A的通量f_x(mol·s^{-1})应与在该截面处浓度梯度的负值成正比：

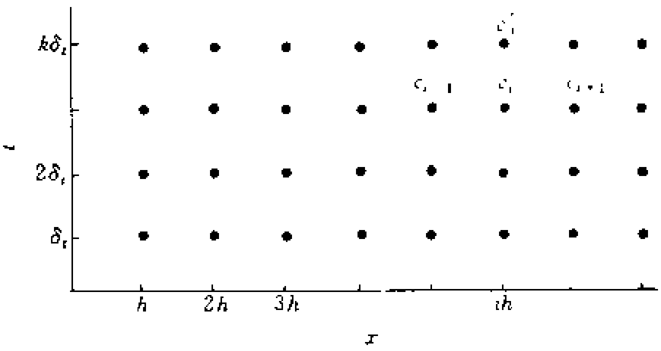


图 8-2　空间与时间离散网点示意图

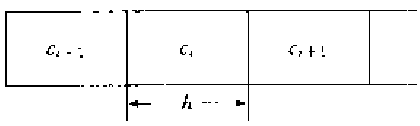


图 8-3　离散的“箱”单元

$$f_x = \mathrm{d}n/\mathrm{d}t = -AD\partial c/\partial x \qquad (8\text{-}28)$$

式中，n 是以摩尔为单位表示的物质的量，D 为扩散系数。

对于图 8-2 所示的“箱”单元，i、$i-1$ 两个单元之间的浓度梯度 $\partial c/\partial x$ 可以 $(c_i-c_{i-1})/h$ 代替，同样的情况适用于 i、$i-1$ 两个单元，故有

$$f_1 = -AD[(c_i-c_{i-1})/h]$$

$$f_2 = -AD[(c_{i+1}-c_i)/h]$$

流入单元 i 的净通量为

$$f = f_1 - f_2$$

故在 δt 时间内，i 单元(体积 V_i)净增加的溶质的量是

$$n = f\delta t$$

$$\delta c_i = n/V_i = (f\delta t)/(Ah) = [(AD\delta t)/(Ah)][(c_{i-1}-2c_i+c_{i+1})/h]$$

$$= [(D\delta t)/(h)][(c_{i-1}-2c_i+c_{i+1})/h]$$

即时间 δt 后，i 单元中浓度为

$$c_i' = c_i + [(D\delta t)/(h^2)](c_{i-1}-2c_i+c_{i+1}) \qquad (8\text{-}29)$$

用 i 单元新的浓度值 c_i'，可进而处理 $i+1$ 单元，如此类推。只是对于直接与电极表面接触的单元，情况稍有不同（即微分方程的边值问题）。设直接与电极接触处浓度为 c_0（参见图 8-4），这里讨论的各单元中的 c_i 均是指其中心点的浓度，而 c_1 与电极-溶液界面处 c_0 之间浓度的梯度为

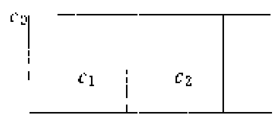


图 8-4　与电极表面接触处的“箱”单元

$$(\partial c/\partial x)_1 = (c_1-c_0)/(h/2)$$

因而

$$f_1 = -AD[(c_1-c_0)/(h/2)]$$

f_2 仍如前表述

$$f_2 = -AD[(c_1-c_0)/h]$$

故有

$$c_1' = c_1 + [(D\delta t)/(h^2)](2c_0-3c_1+c_2) \qquad (8\text{-}30)$$

上面就是“箱”法的最基本部分。这种算法十分直观，体现了在计算机上将溶液离散化逐段模拟的过程。其实，进行这种模拟并不一定需要人为地设想这种“箱”单元，而可直接采用一般的算法。事实上，如有 y 作为 x 的函数（图 8-5），设需要计算等距离的对应于坐标 x、y 的

点 p_{i-1}，p_i，p_{i+1}处一阶微分的近似值，所谓等距离是指 x_{i-1}、x_i、x_{i+1}均相距 h，对于一阶微分（$\mathrm{d}y/\mathrm{d}x$），可写出

$$(\mathrm{d}y/\mathrm{d}x)_{向前} \doteq (y_{i+1}-y_i)/h$$

或

$$(\mathrm{d}y/\mathrm{d}x)_{向后} = (y_i-y_{i-1})/h$$

或

$$(\mathrm{d}y/\mathrm{d}x)_{中心} = (y_{i+1}-y_{i-1})/(2h)$$

对于二阶微分

$$\mathrm{d}^2y/\mathrm{d}x^2 = [(\mathrm{d}y/\mathrm{d}x)_{向前}-(\mathrm{d}y/\mathrm{d}x)_{向后}]/h$$
$$=(y_{i-1}-2y_i+y_{i+1})/h^2$$

现从 Fick 第二定律出发，考察图 8-2 的网格上的三点 c_{i-1}、$2c_i$，c_{i-1}，欲计算 c_i'。将式（8-27）离散化，即以相应差分公式代替微分，左边简写成

$$(\partial c/\partial t)_i = (c_i'-c_i)/(\delta t) \tag{8-31}$$

而对右边则有

$$D(\partial^2 c/\partial x^2) = (D/h^2)(c_{i-1}-2c_i+c_{i+1}) \tag{8-32}$$

合并(8-31)，(8-32)两式，得

$$(c_i'-c_i)/(\delta t) = (D/h^2)(c_{i-1}-2c_i+c_{i+1}) \tag{8-33}$$

亦即

$$c_i' = c_i+[(D\delta t)/h^2](c_{i-1}-2c_i+c_{i+1}) = c_i+\lambda(c_{i-1}-2c_i+c_{i-1}) \tag{8-34}$$

比较式(8-34)与式(8-30)，它们是完全等同的。即不人为想象“箱”单元，可得到相同的结果。

上面通过对最简单的扩散过程的数字模拟，解释了有关算法的基本思路。从这一基本思路出发，可以构造较复杂的系统的模拟算法，如可模拟滴汞电极球形表面的扩散过程等。上述模拟中的 λ，即$[(D\delta t)/h^2]$，是一个很重要的量，对模拟过程十分重要。设计模拟算法时，需控制取 h、δt 的值，使得 $\lambda \leqslant 0.5$。如 $\lambda > 0.5$，模拟将不收敛而出现“振荡”，无法反映扩散过程的实际情况。这可理解为在给定的 D 与 h 下，δt 的选取受到限制，只有在较小的 δt 下才能假定可用线性关系描述扩散过程。如再考察式（8-34），即

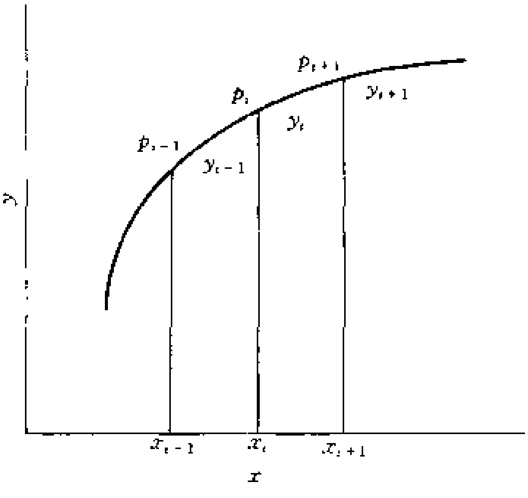


图 8-5 “点”式离散过程

$$c_i' = c_i+\lambda(c_{i-1}-2c_i+c_{i-1})$$

这个式子表述了 c_i' 值是由 c_i 值加右边的另一项而得。这一项中的 c_{i-1}、c_i、c_{i+1}都会带进误差，设这一误差为 e，则有

$$c_i' = c_i+e_i+\lambda(c_{i-1}+e_{i-1}-2c_i-2e_i+c_{i+1}+e_{i+1})$$

如设想对所有 $i-1$、i、$i+1$ 的误差均为 e，则误差大部分抵销，不致因传递而放大。但实际情况难满足这一条件，如设各 e 绝对值相同，符号有正有负，考虑最坏情况，令 c_{i-1}、c_{i+1}的误差为正，而 c_i 的误差为负，则有

$$c_i' = c_i+\lambda(c_{i-1}-2c_i+c_{i+1})+(4\lambda-1)e \tag{8-35}$$

式(8-35)右边的前两项代表 c_i' 的真值，最后一项是模拟过程引入的误差。从式(8-35)可以看

出，$\lambda>0.5$ 时，误差将被放大。

在模拟电化学过程时，常常同一形式的曲线需反复迭代计算。为使这种计算不致于每次由于数值变化要重复进行，可将模拟单位"无因次化"，即将各物理量变换为某种参比值的倍数。例如时间以某种实验观测时间 τ 作参比，变换为

$$T=t/\tau$$

距离则变换为

$$X=x/\delta$$

δ 为 Nernst 扩散层厚度

$$\delta=\sqrt{\pi D\tau}$$

关于 Nernst 扩散层厚度的涵义，这里可设想溶液（本体浓度 c_0）与电极表面接触，接触处发生电化学过程，物质耗尽 $c=0$，则在这个接触处与溶液本体之间将发生浓度差梯度，扩散过程将使本体溶液中物质向电极表面转移，实际上，只在距电极表面几个 δ 厚度处，有这种浓度梯度变化存在，距离更远时浓度均等于本体浓度 c_0。

如将浓度 c 亦无因次化，使之为

$$C=c/c_0$$

代入式(8-27)，将得到

$$\partial C/\partial T=(1/\pi)\partial^2 C/\partial X^2 \tag{8-36}$$

如取 $\delta=\sqrt{D\tau}$，则式(8-36)中的 π 亦可消去，用这样的办法进行模拟，可显著缩短计算机时。

实际的电化学过程有许多因素要加以考虑。除扩散以外，有对流、迁移等过程，有均相的与非均相的化学反应动力因素、吸附动力学因素等。模拟要考虑边界条件，控制电流或电位下的可逆及准可逆体系、多步反应等，条件各不相同，这些体系的数字模拟问题已基本上得到解决。现代计算机数值方法还为电化学过程的模拟提供了工具。进行电化学过程数字模拟最关键的问题是电化学体系的数学描述，可参阅有关电化学及电分析化学专著[24~27]。例如，在作者实验室进行了新合成的电分析试剂的电极过程研究[28]。数字模拟用于电化学研究的另一个实例是离子选择性电极膜电化学过程的模拟，例如中性载体电极响应机理的数字模拟研究。对于新合成制备的基于主-客配合物反应的伯胺电极，用数字模拟方法探讨了各种实验因素对电极性能的影响，并将它与实验结果进行了比较，以阐明有关响应机理[29]。

参 考 文 献

1 Harbaugh J W, Bonham-Carter G. Computer Simulation in Geology. R.E. Krieger Pub Co, Malabar. 1981

2 徐钟济．电子计算机动态，1962：5：45

3 张建中．数学的实践与认识，1974；1：28；1974：2：43

4 李惕培．实验的数学处理．北京：科学出版社，1980

5 刘德贵，费景高，于泳江等．Fortran 算法汇编．北京：国防工业出版社，1983

6 Schaad L J. J Amer Chem Soc. 1963; 85: 3588

7 Manock J J. Spectroscopy and Kinetics. Marcel Dekker, 1973. p.267

8 王柯敏，俞汝勤．第一届全国计算化学会议论文．兰州，1984

9 Gillespie D T. J Comput Phys, 1976; 22: 403

10 梁逸曾，俞汝勤．高等学校化学学报，1988；9：921～925

11 赵学庄．化学动力学原理．上册．北京：高等教育出版社，1985

12 Field R J et al. J Chem Phys. 1974; 60: 1877

13 Efstathiou C E, Hadjiioannou T P. Anal Chem. 1982; 54: 1525

14 Liang Y Z. Yu R Q. Criterion for Judging Number of Principal Factors and Their Evaluation by Monte Carlo Simulation. Paper B-18 of 8th ICCCRE, Beijing: 1987

15 徐钟济. 蒙特卡罗方法. 上海:上海科技出版社,1985.

16 Lambert J D. Computational Methods in ordinary Differential Equations. John Wiley & Sons. 1972: 122

17 Lapidus L, Seinfeld J H. Numerical Solution of Ordinary Differential Equations. New York and London: Academic Press, 1971

18 Hamming R W. J ACM, 1959; 14: 347

19 Gear C W 著. 费景高, 刘德贵, 高永春译. 常微分方程初值问题的数值解法. 北京:科学出版社, 1978

20 梁逸曾, 俞汝勤. 高等学校化学学报, 1988; 9: 881～886

21 Feldberg S W, Auerbach C. Anal Chem, 1964; 36: 505

22 Bard A. J. (Ed). Electroanal Chem. 1969; 3; 199

23 Britz D. Digital Simulation in Electrochemistry. 2nd Ed. Berlin: Springer Verlag. 1988

24 高小霞等. 电分析化学导论. 北京:科学出版社, 1986

25 田昭武. 电化学研究方法. 北京:科学出版社, 1984

26 查全性. 电极过程动力学导论. 北京:科学出版社, 1976

27 高鸿, 张祖训. 极谱电流理论. 北京:科学出版社, 1986

28 张正奇, 俞汝勤. 中国科学 B 辑, 1988; (8): 818

29 刘昆元, 俞汝勤. 中国科学 B 辑, 1990; (3): 283

第九章　化学构效关系的研究方法

定量结构活性关系研究（QSAR）和定量结构性质关系研究（QSPR），这两种定量构效关系研究现已成为化学计量学研究的一个十分重要的分支．化学计量学的发展，将 QSAR 和 QSPR 的研究又提高到了一个新的水平，近年来得到了很快的发展，大量的研究文章已在目前很多重要的国际刊物发表，这些刊物包括美国的“分析化学”（Analytical Chemistry）、《医药化学杂志》（J. Medicinal Chemistry）、《应用毒物学基础》（Fundamentals of Applied Toxicology）、《药物信息杂志》（Drug Information Journal）、《环境健康远景》（Environmental Health Perspectives）、《药物科学杂志》（Journal of Pharmaceutical Science）等，此外，在1982 年一本专门介绍定量构效关系的杂志——Quantitative Structure-Activity Relationship 也已问世。

其实，构效关系的研究可追溯到上个世纪，有机反应性相关分析（CAOC）这一研究领域[1]就可看成是定量构效关系研究的前身。Hammett 在其经典著作[2]中提出线性自由能关系（LFER）可认作 QSAR 和 QSPR 的起点。当时只是少部分的化学参数可以数量化，如取代基的电子效应（Hammett 的 σ 常数），立体参数（Taft 参数）以及疏水性（Hansch 的分配常数）。Hansch[3]的研究也就是将这些化学性质建立在定量研究的基础之上，他主要探讨了结构变化与生化活性之间的关系。我国的化学家们在这方面也做了大量工作，陈荣悌等[4]发展了配位化学中的线性自由能关系的研究。蒋明谦系统论述了有机化合物的同系线性规律[5]，徐光宪等探讨了这类规律的量子化学基础[6]。在有关物理有机化学溶剂效应研究中，刘有成等比较了有关溶剂极性经验参数[7]。定量构效关系的研究实际上是涉及到了化学学科一个带根本性的问题——如何从物质的化学成分与结构定量预测其化学特性？量子化学研究的一个重要目的就是要直接从微观的角度来回答这样的问题。对于一个给定的分子，量子化学通过求解相应的 Schrodinger 方程，求出相应的特征值和特征向量，可描绘能量和电子的分布，分子的性质可从波函数与能量导出。目前，量子化学已取得令人瞩目的成就，但对于复杂的分子进行量子化学计算尚有一定困难。由于这一原因，大量的化学研究仍处于一种“定性”的水平。而本书所主要论及的定量构效关系则是从宏观的角度出发，直接从试验数据或某些量化的结构数据出发，采用统计学和模式识别的方法来建立起某些化学结构与性能的关系。

因有机反应性相关分析这一研究领域实际可看成是定量构效关系研究的前身，所以本章首先将对有机反应性相关分析这一研究领域进行必要的介绍，然后，对化合物结构的数字表征方法，如拓扑矩阵、分子连接性指数等，作出详细地介绍，在此基础上，再对化学计量学中的构效关系建模方法分成三个部分，即基于回归分析的建模方法、基于模式识别的建模方法和人工神经网络的建模方法分别给出详细说明。

第一节　有机反应性相关分析方法简介

有机反应性相关分析（CAOC）这一研究领域主要应用回归分析方法来寻找有机反应性的数量关系。历史上，线性自由能关系（LFER）就是由有机化学家发展起来的，同时他们还提出一些有用的反应参数，如取代基的电子效应（Hammett 的 σ 常数），立体参数（Taft 等的参

数）等，本节对这些内容将给出简要介绍，以保持定量构效关系这一内容的完整性。

一、线性自由能概念及其经典 Hammett σ 常数

对于热力学平衡常数 K，可写出

$$\lg K=\frac{-\Delta G^{\circ}}{2.303RT}$$

ΔG° 是反应的标准自由能变化，对动力学过程，速度常数 k_r 可表示为

$$\lg k_r=\lg\frac{kT}{h}-\frac{\Delta G^{\ne}}{2.303RT}$$

此方程中 k 是 Boltzman 常数，$\Delta G^{\ne}$ 是标准活化自由能。由此得出，有关 $\lg K$ 或 $\lg k$（恒定温度下）的关系，实质上是关于自由能的关系，故有线性自由能（LFER）之称。

Hammett[2]提出的经典公式，描述了间位或对位取代苯衍生物的支链反应的速度常数 k 或平衡常数 K 与取代常数 σ、反应常数 ρ 的关系：

$$\lg k_r=\lg k_r^{\circ}+\rho\sigma$$

$$\lg K=\lg K^{\circ}+\rho\sigma$$

k_r°、K° 是与未取代的母化合物性质有关的常数，取代常数 σ 表征的是将苯环上的氢以给定的间位或对位取代基取代时的极性效应，原则上与温度、溶剂介质等条件无关。而反应常数 ρ 则与这些条件有关。Hammett 选择苯甲酸在 25℃ 时在水中的电离作为参比反应，定义 $\rho=1$，并定义

$$\sigma=\lg(K_a/K_a^{\circ})\tag{9-1}$$

式中，K_a° 是苯甲酸本身的离解常数，K_a 则是取代苯甲酸的离解常数，$\lg k_r$ 或 $\lg K$ 与 σ 之间有线性关系，这就是经典的 LFER。

不同基团在间位（m）或对位（p）的 σ 值示例于表 9-1 [8]。符合于方程（9-1）的构效关系已观察到差不多有数百种[9]。平衡常数和速度常数都符合于这种关系。值得注意的是并非所有的化合物系列在 σ 和 $\lg k$ 或 $\lg K$ 间均能获得简单的线性关系。此外，尽管上述关系是用热力学术语加以叙述的（ΔG 或 $\lg K$ 值），但只能属于超热力学范畴，因为无法用热力学原理来阐述这些关系是真正存在的。

表 9-1　一些取代基的 σ 值

取代值	σ_m	σ_p	取代值	σ_m	σ_p
Me	−0.07	−0.16	OH	0.13	−0.38
Bu	−0.10	−0.20	OMe	0.11	−0.28
Ph	0.06	−0.01	OAC	0.39	0.31
CF_3	0.43	0.54	SH	0.25	0.15
COMe	0.38	0.50	SMe	0.15	0.00
CO_2H	0.37	0.45	$SO_2\cdot Me$	0.60	0.72
CN	0.61	0.70	F	0.34	0.06
NH_2	0.00	−0.57	Cl	0.37	0.22
NMe_2	−0.15	0.63	Br	0.39	0.22
NO_2	0.71	0.78	I	0.35	0.21

二、取代基电子效应常数（σ 常数）的应用与拓展

取代基的电子效应常数（σ 常数）对线性自由能类型的构效关系分析很有用，它主要是为了说明各种远程取代基对分子某部分的反应性能的影响。

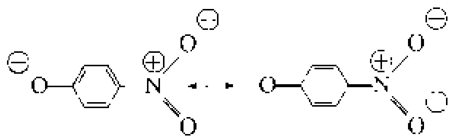
（一）芳香系统上的取代基

传统的 Hammett 方程由式（9-1）给出，并定义了苯甲酸在 25℃ 时在水中的电离的 r 等于 1，由此可算出其他一些反应的 σ 值（表 9-2）。表的上半部分表示在芳环和反应中心之间的次甲基越多，σ 值就越小（σ 值衡量反应对取代基影响的敏感程度）。表的下部分表明溶剂的极性变小，导致 σ 值增大，这种溶剂效应尚未充分认识[11]。

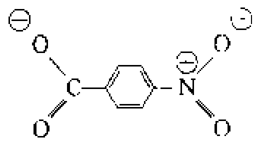
表 9-2　羧酸反应中溶剂和侧链对 σ 值的影响

底　物	溶　剂	温度/℃	σ 值	底　物	溶　剂	温度/℃	σ 值
酸的电离作用				ArCOOH	异丙醇	30	1.07
ArCOOH	水	25	1.00	ArCOOH	乙醇	30	0.94
$ArCH_2COOH$	水	25	0.49	ArCOOH	甲醇	30	0.88
ArCH=CHCOOH(反式)	水	25	0.47	ArCH=CHCOOH(反式)	乙醇	30	0.42
$ArSCH_2COOH$	水	25	0.30	$ArCH_2COOH$	乙醇	30	0.40
$ArSO_2CH_2COOH$	水	25	0.25	$ArOCH_2COOH$	乙醇	30	0.25
$Ar(CH_2)_2COOH$	水	25	0.21	$Ar(CH_2)_2COOH$	乙醇	30	0.22
酸与偶氮二苯甲烷的反应				p-ArC_6H_4COOH	乙醇	30	0.22
ArCOOH	叔丁醇	30	1.28				

在 Hammett 方程式定义后不久，发现某些基团或母体分子或反应需要采用特殊的 σ 值。这种 σ 值反映了取代基和反应部位间直接共轭相互作用的影响。对位硝基使酚盐稳定化作用就是个典型例子（XXII），然而，这种共轭稳定化在对硝基苯甲酸盐中是不存在的（XXII）。



XXII



XXII

所以在取代基和反应中心有直接共轭作用时，需另用一类 σ 值。凡在反应时毗邻环处产生正性中心时，用 σ_p^- 表示对位取代基的电子常数，例如，$-OCH_3$，$-CH_3$，$-OH$，$-SCH_3$，它们通过直接共轭向正性中心供给电子；反之，可产生负性中心的反应，用 σ_p^- 表示取代基的电子常数，例如，对位取代的$-NO_2$，$-CN$，$-COOH$，$-SONH_2$。其他取代基仍用原值表示。表 9-3 列出了常用取代基的 σ，σ_p^+，σ_p^- 值。

表 9-3　常用取代基的电子参数

取代基	σ_m	σ_p	σ_p^+	σ_p^-	取代基	σ_m	σ_p	σ_p^+	σ_p^-
H	0.00	0.00	0.00	0.00	SO_2CH_3	0.68	0.73		1.05
F	0.34	0.15	−0.07	−0.02	SCH_3	0.14	0.06	−0.60	
Cl	0.47	0.23	0.11		$COCH_3$	0.36	0.47		0.87
Br	0.37	0.26	0.15		CO_2R	0.35	0.44	0.48	0.68
Me	−0.06	−0.14	−0.31		C_6H_5	0.06	−0.05	−0.17	
NH_2	−0.09	−0.30	−1.30		OC_6H_5	0.25	−0.06	−0.50	
OH	0.02	−0.22	−0.92		$CONH_2$	0.28	0.36		0.62
$N(CH_3)_3^+$	0.88	0.82	0.41[b]		NO_2	0.71	0.81	0.79	1.24
$N(CH_3)_2$	−0.10	−0.32	−1.70		CN	0.62	0.71	0.66	0.90
OCH_3	0.10	−0.12	−0.78	−0.20	CF_3	0.46	0.54		

应用这些特殊的 σ 常数所遇到的问题是：有三种尺度而不是一种。在实际计算中，不可能直截了当地决定用哪种，物理有机化学家对此有所考虑，许多反应需要采用正常的和修正值之间的 σ 值。

（二）σ 值分成共振成分和诱导成分

评价共轭效应在反应中的重要性的一个方法就是分别应用取代基的共轭效应常数（或称共振效应常数）和诱导效应常数。Taft 和 Lewis[10] 1958 年提出 σ 值是这两种效应的线性组合。根据化学原理，他们创立了 σ_I（诱导）和 σ_R（共振）值表。这样，无论对于哪类 σ 值，只是下式中的 a，b 系数不同：

$$\sigma = a\sigma_I + b\sigma_R$$

三、取代基的立体效应常数

在选用各种参数时，立体效应参数是最成问题的。这是因为某化合物或反应的立体相互作用难以平移到另一化合物或反应中，而且，识别哪种立体影响是重要的也很不容易。下面介绍两种立体效应常数。

（一）Taft 的立体参数

基团立体效应的经典线性自由能度量是 Taft[10]的立体参数 E_s 值。E_s 的定义是：在相同的溶剂、温度、酸度下，酰基取代的乙酸甲酯的酸催化水解相对速度与乙酸甲酯的比值的对数

$$E_{sx} = \lg K_{xCO_2Me} - \lg K_{MeCO_2Me}$$

式中，K_{xCO_2Me}是被取代的酯的水解速度常数；K_{MeCO_2Me}是乙酸甲酯的水解速度常数。因此，甲基是基准取代基，若以 K 和 K_0 分别表示取代的和基准化合物的速度常数，则另一种表达式如下：

$$E_s = \lg|K/K_0|_a \tag{9-2}$$

式中，下标 a 指酸水解反应。

酸催化水解反应速度是否仅由立体效应控制，现在仍有不少争论。通过观测发现如下基本事实：苯甲酸酯间位或对位取代基几乎不影响酸催化水解反应速度；苯甲酸上类似取代基也不影响酸催化水解反应速度。而这类取代基对同样化合物的碱催化反应速度影响却很大。

Hancock 等[12]认为该模型是受超共轭效应影响的，建议 E_s 应按（9-3）式加以修正：

$$E_s^0 = E_s + 0.306(n-3) \tag{9-3}$$

式中，n 是 α-氢原子的数目。这一论断已被量子化学计算证实。

许多取代基的 TaftE_s 值是不能测量的，系因 E_s 值是依模型反应的性质所定。针对这个问题，Charton[13]和 Kutter 及 Hansch[14]建议，对于球形对称的基团，可用半径作为立体效应的度量。E_s 和半径间的良好相关性证实了 E_s 的立体性质。利用这种关系，已经计算了另外一些基团的 E_s 值。

（二）Verloop 立体参数

Verloop 等[15,16]介绍了一套新的立体取代基参数。它们是通过标准键角和键距的计算而得到的。已经发表了 243 个取代基的 5 个参数，第一个参数为 L，它是沿取代基和母体分子的键轴方向上的取代基长度。垂直于该键轴方向又量测出 4 个参数 B_1～B_4，B_1 是最小宽度参数，B_2～B_4 分别是垂直于其他 B 的距离，且依值的大小递增排列。故 L 和 B_1～B_4 反映了紧紧围绕基团的 5 个平面的位置，这些位置与连接点和键轴有关。

从初步应用看来，这些参数在构效关系的计算中是非常有用的。但是，是否 5 个参数都

必要，尚未作定论。表 9-4 引摘了 243 个取代基的 Verloop 参数供读者参考。

表 9-4 243 个取代基的 Verloop 立体参数

编 号	取 代 基	L/nm	B_1/nm	B_2/nm	B_3/nm	B_4/nm
1	H	0.206	0.100	0.100	0.100	0.100
2	F	0.265	0.135	0.135	0.135	0.135
3	Cl	0.352	0.180	0.180	0.180	0.180
4	Br	0.383	0.195	0.195	0.195	0.195
5	I	0.423	0.215	0.215	0.215	0.215
6	CH_3	0.300	0.152	0.190	0.190	0.204
7	C_2H_5	0.411	0.152	0.190	0.190	0.297
8	n-C_3H_7	0.505	0.152	0.190	0.190	0.349
9	i-C_3H_7	0.411	0.204	0.276	0.316	0.316
10	c-C_3H_5	0.414	0.198	0.224	0.229	0.288
11	n-C_4H_9	0.617	0.152	0.190	0.190	0.442
12	i-C_4H_9	0.505	0.152	0.190	0.316	0.421
13	s-C_4H_9	0.505	0.190	0.276	0.316	0.349
14	t-C_4H_9	0.411	0.259	0.286	0.286	0.297
15	c-C_4H_7	0.469	0.195	0.264	0.264	0.383
16	n-C_5H_{11}	0.711	0.152	0.190	0.190	0.494
17	i-C_5H_{11}	0.617	0.152	0.190	0.316	0.442
18	$CH(C_2H_5)_2$	0.473	0.211	0.259	0.399	0.399
19	c-C_5H_9	0.497	0.204	0.286	0.286	0.398
20	n-C_6H_{13}	0.822	0.152	0.190	0.190	0.587
21	$C_2H_4C(CH_3)_3$	0.617	0.152	0.286	0.286	0.442
22	$CH_2C(CH_3)_2C_2H_5$	0.617	0.152	0.316	0.316	0.442
23	c-C_6H_{11}	0.617	0.204	0.316	0.316	0.349
24	n-C_7H_{15}	0.916	0.152	0.190	0.190	0.639
25	$C(CH_3)_2C(CH_3)_3$	0.505	0.259	0.312	0.312	0.414
26	n-C_8H_{17}	1.027	0.152	0.190	0.190	0.733
27	n-C_8H_{17}(紧密型)	0.582	0.152	0.217	0.475	0.685
28	n-$C_{12}H_{25}$	1.438	0.152	0.190	0.190	1.027
29	n-$C_{12}H_{25}$(紧密型)	0.729	0.152	0.190	0.466	0.939
30	CH_2F	0.330	0.152	0.190	0.190	0.261
31	CH_2Cl	0.389	0.152	0.190	0.190	0.346
32	CH_2Br	0.409	0.152	0.195	0.195	0.375
33	CH_2I	0.436	0.152	0.215	0.215	0.415
34	CH_2OH	0.397	0.152	0.190	0.190	0.270
35	CH_2OCH_3	0.491	0.152	0.190	0.190	0.288
36	CH_2OCOCH_3	0.546	0.152	0.190	0.190	0.446
37	$CH_2OC_6H_5$	0.819	0.152	0.309	0.311	0.311
38	CH_2SCH_3	0.541	0.152	0.190	0.190	0.341
39	$CH_2SO_2CH_3$	0.505	0.152	0.252	0.252	0.348
40	CH_2NHCH_3	0.496	0.152	0.190	0.190	0.290
41	$CH_2N(CH_3)_2$	0.483	0.152	0.280	0.280	0.342
42	$CH_2\overset{+}{N}(CH_3)_3$	0.496	0.152	0.280	0.280	0.394
43	CH_2NO_2	0.370	0.152	0.244	0.244	0.336
44	CH_2CCH	0.399	0.152	0.190	0.190	0.449
45	CH_2CN	0.399	0.152	0.190	0.190	0.412
46	CH_2CHCH_2	0.511	0.152	0.190	0.190	0.378
47	CH_2CBrCH_2	0.416	0.152	0.312	0.368	0.413

续表

编 号	取 代 基	L/nm	B_1/nm	B_2/nm	B_3/nm	B_4/nm
48	CH_2CF_3	0.470	0.152	0.244	0.244	0.345
49	CH_2COCH_3	0.454	0.152	0.190	0.190	0.439
50	CH_2COOCH_3	0.598	0.152	0.190	0.190	0.428
51	$CH_2C_6H_5$	0.363	0.152	0.311	0.311	0.602
52	$CH_2(p\text{-}ClC_6H_4)$	0.442	0.152	0.311	0.311	0.744
53	$CH_2CONHC_6H_5$	0.695	0.152	0.311	0.311	0.722
54	C_2H_4Cl	0.557	0.152	0.190	0.190	0.325
55	C_2H_4Br	0.587	0.152	0.195	0.340	0.340
56	C_2H_4I	0.628	0.152	0.215	0.215	0.360
57	C_2H_4OH	0.479	0.152	0.190	0.190	0.338
58	$C_2H_4NO_2$	0.549	0.152	0.244	0.244	0.298
59	$C_2H_4N(CH_3)_2$	0.558	0.152	0.280	0.280	0.297
60	C_2H_4CN	0.628	0.152	0.190	0.190	0.305
61	C_2H_4COOH	0.596	0.152	0.235	0.267	0.305
62	$C_2H_4CONH_2$	0.611	0.152	0.242	0.297	0.308
63	$C_2H_4C_6H_5$	0.833	0.152	0.311	0.311	0.316
64	CHF_2	0.330	0.171	0.175	0.226	0.256
65	$CHCl_2$	0.389	0.204	0.263	0.324	0.324
66	$CHBr_2$	0.409	0.204	0.285	0.350	0.350
67	$CHFCl$	0.389	0.175	0.228	0.258	0.339
68	$CHFBr$	0.409	0.175	0.247	0.258	0.368
69	$CHClBr$	0.409	0.190	0.275	0.346	0.356
70	$CHClCH_3$	0.389	0.189	0.258	0.287	0.346
71	$CHBrCH_3$	0.409	0.193	0.283	0.298	0.352
72	$CHOHCH_3$	0.397	0.176	0.222	0.263	0.309
73	$CHCH_3N(CH_3)_2$	0.429	0.190	0.305	0.374	0.424
74	$CH(C_6H_5)_2$	0.515	0.201	0.499	0.528	0.582
75	CF_3	0.330	0.198	0.244	0.244	0.261
76	CCl_3	0.389	0.263	0.324	0.324	0.346
77	CBr_3	0.409	0.286	0.350	0.350	0.375
78	CF_2CF_3	0.411	0.198	0.244	0.244	0.364
79	$C(CH_3)_2CH_2OCH_3$	0.555	0.277	0.316	0.316	0.449
80	$C(C_2H_5)_2C_6H_5$	0.528	0.310	0.332	0.390	0.596
81	$C(C_6H_5)_3$	0.515	0.484	0.502	0.571	0.594
82	$CHCH_2$	0.429	0.160	0.160	0.200	0.309
83	$CHCHCH_3(E)$	0.523	0.190	0.190	0.200	0.309
84	$CHCHCH_3(Z)$	0.429	0.190	0.190	0.200	0.403
85	$C(CH_3)CH_2$	0.429	0.190	0.190	0.309	0.311
86	$CHCHCH(CH_3)_2$	0.635	0.190	0.200	0.316	0.406
87	$CHCHNO_2$	0.429	0.160	0.183	0.324	0.421
88	$CHCHCOOH$	0.615	0.160	0.183	0.251	0.311
89	$CHCHCOCH_3$	0.580	0.160	0.183	0.324	0.373
90	$CHNOH(E)$	0.488	0.160	0.160	0.192	0.311
91	$CHNOH(Z)$	0.394	0.160	0.183	0.270	0.356
92	$CHNCH(CH_3)_2$	0.635	0.190	0.200	0.316	0.406
93	$CHNN(CH_3)_2$	0.577	0.190	0.200	0.280	0.417
94	$CHNC_6H_5$	0.850	0.170	0.170	0.236	0.407
95	COH	0.353	0.160	0.160	0.200	0.236
96	$COCH_3$	0.406	0.190	0.190	0.236	0.293

续表

编号	取代基	L/nm	B_1/nm	B_2/nm	B_3/nm	B_4/nm
97	$COCH(CH_3)_2$	0.467	0.236	0.316	0.316	0.369
98	COC_6H_5	0.457	0.236	0.311	0.311	0.598
99	COOH	0.391	0.160	0.160	0.236	0.266
100	$COOCH_3$	0.485	0.190	0.190	0.236	0.336
101	$COOC_2H_5$	0.596	0.190	0.190	0.236	0.429
102	$COOC_3H_7$	0.690	0.190	0.190	0.236	0.483
103	$CONH_2$	0.406	0.160	0.160	0.242	0.307
104	$CONHCH_3$	0.500	0.160	0.223	0.242	0.307
105	$CONHNH_2$	0.492	0.160	0.195	0.242	0.328
106	$CON(CH_3)_2$	0.461	0.190	0.218	0.353	0.383
107	$CONCH_3CONHCH_3$	0.461	0.181	0.225	0.352	0.543
108	CCH	0.466	0.160	0.160	0.160	0.160
109	CCC_6H_5	0.888	0.170	0.170	0.311	0.311
110	CN	0.423	0.160	0.160	0.160	0.160
111	C_6H_5	0.628	0.170	0.170	0.311	0.311
112	p-ClC_6H_4	0.774	0.180	0.180	0.311	0.311
113	OH	0.274	0.135	0.135	0.135	0.193
114	OCH_3	0.398	0.135	0.190	0.190	0.287
115	OC_2H_5	0.492	0.135	0.190	0.190	0.336
116	n-OC_3H_7	0.605	0.135	0.190	0.190	0.430
117	i-OC_3H_7	0.459	0.135	0.190	0.316	0.361
118	c-OC_3H_5	0.468	0.135	0.224	0.229	0.359
119	n-OC_4H_9	0.699	0.135	0.190	0.190	0.479
120	i-OC_4H_9	0.605	0.135	0.190	0.316	0.430
121	s-OC_4H_9	0.605	0.135	0.190	0.316	0.430
122	t-OC_4H_9	0.492	0.135	0.286	0.286	0.394
123	n-OC_5H_{11}	0.811	0.135	0.190	0.190	0.573
124	c-OC_5H_9	0.602	0.135	0.286	0.290	0.400
125	n-OC_6H_{13}	0.905	0.135	0.190	0.190	0.623
126	c-OC_6H_{11}	0.597	0.135	0.316	0.316	0.529
127	n-OC_7H_{15}	1.018	0.135	0.190	0.190	0.716
128	n-OC_8H_{17}	1.112	0.135	0.190	0.190	0.766
129	n-OC_8H_{17}(紧密型)	0.875	0.135	0.217	0.442	0.475
130	OCH_2OH	0.466	0.135	0.190	0.190	0.326
131	OCH_2CCH	0.658	0.135	0.190	0.190	0.293
132	$OCH_2C_6H_5$	0.820	0.135	0.303	0.311	0.311
133	$OC_2H_4OCH_3$	0.684	0.135	0.190	0.190	0.471
134	OCF_3	0.457	0.135	0.244	0.244	0.333
135	$OCCl_3$	0.544	0.135	0.324	0.324	0.398
136	$OCHCH_2$	0.498	0.135	0.160	0.160	0.365
137	$OCHCHCH_3(E)$	0.528	0.135	0.190	0.190	0.477
138	$OCHCHCH_3(Z)$	0.593	0.135	0.190	0.190	0.365
139	$OCCH_2CH_3$	0.487	0.135	0.190	0.190	0.440
140	OCOH	0.393	0.135	0.160	0.160	0.368
141	$OCOCH_3$	0.487	0.135	0.190	0.190	0.368
142	$OCOC_6H_5$	0.815	0.170	0.170	0.184	0.440
143	OCOOH	0.471	0.135	0.160	0.160	0.368
144	$OCONH_2$	0.482	0.135	0.160	0.160	0.362
145	$OCONHCH_3$	0.551	0.135	0.190	0.190	0.455

续表

编 号	取 代 基	L/nm	B/nm	B_2/nm	B_3/nm	B_4/nm
146	OCN	0.387	0.135	0.160	0.160	0.401
147	OC_6H_5	0.451	0.135	0.311	0.311	0.589
148	OSO_2CH_3	0.403	0.135	0.190	0.357	0.386
149	$OSO_2C_6H_5$	0.820	0.161	0.180	0.357	0.364
150	SH	0.347	0.170	0.170	0.170	0.233
151	SCH_3	0.430	0.170	0.190	0.190	0.326
152	SC_2H_5	0.524	0.170	0.190	0.190	0.397
153	*n*-SC_3H_7	0.621	0.170	0.190	0.190	0.490
154	*i*-SC_3H_7	0.495	0.170	0.190	0.316	0.415
155	*c*-SC_3H_5	0.504	0.170	0.224	0.229	0.414
156	*n*-SC_4H_9	0.715	0.170	0.190	0.190	0.561
157	*i*-SC_4H_9	0.642	0.170	0.190	0.316	0.490
158	*s*-SC_4H_9	0.621	0.170	0.190	0.316	0.490
159	*t*-SC_4H_9	0.524	0.170	0.286	0.286	0.423
160	*n*-SC_5H_{11}	0.812	0.170	0.190	0.190	0.654
161	*n*-SC_6H_{13}	0.905	0.170	0.190	0.190	0.725
162	*c*-SC_6H_{11}	0.616	0.170	0.316	0.316	0.596
163	*n*-SC_7H_{15}	1.002	0.170	0.190	0.190	0.818
164	*n*-SC_8H_{17}	1.096	0.170	0.190	0.190	0.889
165	*n*-SC_8H_{17}	0.915	0.170	0.217	0.475	0.524
166	SCH_2OH	0.506	0.170	0.190	0.190	0.326
167	SCH_2CCH	0.689	0.170	0.190	0.190	0.361
168	$SCH_2C_6H_5$	0.850	0.170	0.311	0.311	0.386
169	SCF_3	0.489	0.170	0.244	0.244	0.369
170	$SCHCH_2$	0.533	0.170	0.170	0.170	0.423
171	$SCHCHCH_3$(*E*)	0.540	0.170	0.190	0.190	0.526
172	$SCHCHCH_3$(*Z*)	0.629	0.170	0.190	0.190	0.423
173	$SCCH_3CH_2$	0.519	0.170	0.190	0.190	0.476
174	SCOH	0.425	0.170	0.170	0.170	0.401
175	$SCOCH_3$	0.519	0.170	0.190	0.190	0.401
176	$SCONH_2$	0.512	0.170	0.184	0.184	0.401
177	$SCONHCH_3$	0.615	0.170	0.190	0.190	0.488
178	SCN	0.408	0.170	0.170	0.170	0.445
179	SC_6H_5	0.457	0.170	0.311	0.311	0.642
180	$^+S(CH_3)_2$	0.362	0.170	0.259	0.286	0.286
181	$SOCH_3$	0.403	0.160	0.249	0.293	0.336
182	SO_2CH_3	0.437	0.211	0.267	0.267	0.315
183	$SO_2C_2H_5$	0.531	0.211	0.267	0.267	0.367
184	SO_2F	0.350	0.203	0.245	0.251	0.270
185	SO_2CF_3	0.411	0.211	0.267	0.267	0.364
186	$SO_2C_6H_5$	0.586	0.211	0.267	0.267	0.601
187	SO_2NH_2	0.382	0.211	0.267	0.267	0.307
188	$SO_2N(CH_3)_2$	0.508	0.211	0.280	0.280	0.360
189	SO_3^+	0.350	0.211	0.267	0.267	0.270
190	SF_5	0.465	0.249	0.249	0.249	0.249
191	NH_2	0.293	0.150	0.150	0.184	0.184
192	$NHCN_3$	0.353	0.150	0.190	0.190	0.308
193	NHC_2H_5	0.496	0.150	0.190	0.190	0.342
194	*n*-NHC_3H_7	0.607	0.150	0.190	0.190	0.436

续表

编号	取代基	L/nm	B_1/nm	B_2/nm	B_3/nm	B_4/nm
195	i-NHC_3H_7	0.496	0.150	0.190	0.316	0.415
196	c-NHC_3H_5	0.472	0.150	0.224	0.229	0.364
197	n-NHC_4H_9	0.701	0.150	0.190	0.190	0.497
198	i-NHC_4H_9	0.607	0.150	0.190	0.316	0.436
199	s-NHC_4H_9	0.607	0.150	0.190	0.316	0.436
200	t-NHC_4H_9	0.496	0.150	0.286	0.286	0.398
201	n-NHC_5H_{11}	0.813	0.150	0.190	0.190	0.518
202	c-NHC_5H_9	0.606	0.150	0.286	0.290	0.407
203	n-NHC_6H_{13}	0.907	0.150	0.190	0.190	0.633
204	c-NHC_6H_{11}	0.600	0.150	0.316	0.316	0.536
205	n-NHC_7H_{15}	1.018	0.150	0.190	0.190	0.726
206	n-NHC_8H_{17}	1.112	0.150	0.190	0.190	0.778
207	c-NHC_8H_{15}	0.880	0.176	0.217	0.450	0.475
208	$NHCH_2OH$	0.470	0.150	0.190	0.190	0.332
209	$NHCH_2CCH$	0.662	0.150	0.190	0.190	0.299
210	$NHCH_2C_6H_5$	0.824	0.150	0.309	0.311	0.311
211	$NHCHCH_2$	0.502	0.150	0.160	0.184	0.371
212	$NHCHCHCH_3(E)$	0.531	0.150	0.190	0.190	0.482
213	$NHCHCHCH_3(Z)$	0.597	0.150	0.190	0.190	0.371
214	$NHCCH_3CH_2$	0.491	0.150	0.190	0.190	0.444
215	NHCOH	0.422	0.150	0.150	0.194	0.361
216	$NHCOCH_3$	0.515	0.150	0.190	0.194	0.361
217	NHCOOH	0.501	0.150	0.150	0.194	0.361
218	$NHCONH_2$	0.509	0.184	0.184	0.194	0.361
219	$NHCONHCH_3$	0.589	0.184	0.194	0.280	0.361
220	$NHCOOC_2H_5$	0.445	0.150	0.190	0.497	0.557
221	$NHCOC_6H_5$	0.840	0.194	0.311	0.311	0.361
222	$NHCSNH_2$	0.462	0.150	0.234	0.257	0.418
223	NHCN	0.390	0.150	0.160	0.184	0.405
224	NHC_6H_5	0.453	0.150	0.311	0.311	0.595
225	$NHNH_2$	0.340	0.150	0.184	0.184	0.282
226	$NHSO_3CH_3$	0.406	0.150	0.190	0.359	0.388
227	$N(CH_3)_2$	0.353	0.150	0.256	0.280	0.280
228	$N(CF_3)_2$	0.401	0.150	0.310	0.328	0.338
229	$N(C_6H_5)_2$	0.577	0.150	0.464	0.534	0.586
230	NH_3^+	0.293	0.150	0.184	0.184	0.197
231	$\overset{+}{N}(CH_3)_3$	0.402	0.256	0.280	0.280	0.290
232	$NCCl_2$	0.565	0.170	0.180	0.184	0.454
233	NCS	0.429	0.150	0.164	0.176	0.424
234	$NCHC_6H_5$	0.840	0.170	0.180	0.355	0.366
235	NNC_6H_5	0.843	0.170	0.170	0.192	0.431
236	$NCHN(CH_3)_2$	0.577	0.160	0.190	0.280	0.417
237	NO	0.344	0.170	0.170	0.170	0.244
238	NO_2	0.344	0.170	0.170	0.244	0.244
239	N_3	0.462	0.150	0.234	0.257	0.418
240	$P(CH_3)_2$	0.388	0.200	0.284	0.297	0.329

续表

编号	取代基	L/nm	B_1/nm	B_2/nm	B_3/nm	B_4/nm
241	$PO(C_6H_5)_2$	0.540	0.268	0.468	0.548	0.619
242	IO_2	0.425	0.215	0.246	0.338	0.336
243	2-噻吩基	0.597	0.165	0.177	0.313	0.316

四、取代基电子效应常数与立体参数的协同效应

以上讨论一般都是将电子效应和立体效应单独进行考虑的。然而，在实际上有些反应却同时存在着两种效应。所以更一般的情况似应由下式给出：

$$\lg(K/K^{\circ})=\rho\sigma+\delta E_s \tag{9-4}$$

实际上，式(9-4)就隐含着活性自由能是相互独立的电子效应和立体效应的综合结果，用数学语言来说，活性自由能是电子效应和立体效应的线性组合。这样的表达式对平衡常数也同样适合。式（9-4）构成了一个多元线性回归在物理有机化学应用的一个最早的例子，它首先由 Pavelich 和 Taft[17,18]引入定量构效关系研究。他们在对 30℃ 甲醇中酸碱催化(－)-甲基酯 $RCO_2C_{10}H_{19}$的甲醇化的研究中发现，对于酸催化反应，可得 $\rho=0.626$，$\delta=1.549$，复相关系数 $R=0.992$；而对于碱催化反应，可得 $\rho=2.702$，$\delta=1.301$，复相关系数 $R=0.996$。随后 Biechler 和 Taft 又将此关系应用到复杂的 N-甲基苯胺，PhNMeCOR 的碱水解反应动力学研究之中[19]。另一个应用是由 Bowden、Chapman 和 Shorter 设计的，他们发现，具有大基团（如 Ph_2CH，Ph_3C）的芳基酸的 $\lg K_a$ 值单独都与 σ 或 E_s 没有线性关系，然而，如采用由式（9-4）所示的多元线性关系式来进行回归，则得到很满意的结果，他们对在 25℃ 下，50%2-n-丁氧基-乙醇水溶液中的 13 个酸的研究结果可由下面的方程给出

$$\lg K_a=-6.045+2.666\sigma+0.252E_s$$

所得的复相关系数 R 近似为 1[20]。类似这样的例子还不少，读者可参阅文献[1]。

第二节 化合物结构的数量表征方法

化合物的结构多用图来表示，通常这样的图称为结构图或分子图。不同的化合物其结构图也不同，于是，如能将不同的化学结构图数量化，并通过它将结构与活性或性能联系起来，这就是将在这一节将要介绍的分子拓扑指数方法。最早引入此方法是 Wiener[21]。实际上，在 1874 年数学家 Sylvester 就指出，一个化学图可用矩阵表示，这个矩阵叫拓扑矩阵或邻接矩阵，如化合物 2,3-二甲基戊烷（Ⅰ）和 1,1,2-三甲基环丙烷（Ⅱ），它们的结构图与隐氢图示于图 9-1。

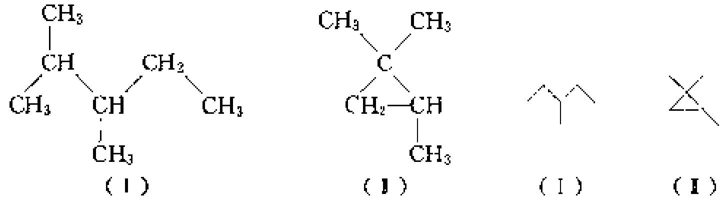


图 9-1 化合物 2,3-二甲基戊烷（Ⅰ）和 1,1,2-三甲基环丙烷（Ⅱ）的结构图与隐氢图

可分别用邻接矩阵 $\boldsymbol{A}_{Ⅰ}$ 和 $\boldsymbol{A}_{Ⅱ}$ 表示，此矩阵中的元素 a_{ij}为

$$a_{ij}=\begin{cases}0 & \text{原子 } i \text{ 和 } j \text{ 没有键合时}\\ 1 & \text{原子 } i \text{ 和 } j \text{ 有键合时}\end{cases}$$

对于（Ⅰ）和（Ⅱ）这两个化合物的邻接矩阵分别为

$$A_1=\begin{pmatrix}0&1&0&0&0&0&0\\1&0&1&0&0&1&0\\0&1&0&1&0&0&1\\0&0&1&0&1&0&0\\0&0&0&1&0&0&0\\0&1&0&0&0&0&0\\0&0&1&0&0&0&0\end{pmatrix}\qquad A_1=\begin{pmatrix}0&0&1&0&0&0\\0&0&1&0&0&0\\1&1&0&1&1&0\\0&0&1&0&1&0\\0&0&1&1&0&1\\0&0&0&0&1&0\end{pmatrix}$$

现以化合物2,3-二甲基戊烷（Ⅰ）为例来加以具体说明。从图9-1可以看出，它的（1,2)、(2,3)、(2,6)、(3,7)、(3,4)、(4,5) 结点之间有边，故矩阵的对应行与对应列的元素是1。这种数字表示法可以把化合物的分子经适当变换存入计算机中。可以看出上述矩阵是以主对角线为轴的对称矩阵，矩阵中的每一行或每一列的元素之和即为分子图中相应编号的那个碳原子的点价。这样，化学结构就可用拓扑矩阵中的具体数字来描述。分子图中的点价为

$$\delta_i=\sum^{n}a_{ij}$$

式中，n 点数即矩阵的阶。就是在这拓扑邻接矩阵的基础上，化学家们提出了很多种分子拓扑指数，据 Trinajstic 等于1986年在数学化学杂志上的评述[22]，在当时就至少有39种不同的拓扑指数，而根据巴纳班 (Balaban) 在一篇评论中指出[23]，大约存在一百多种分子拓扑指数。在本节只将对应用最多的和新近发展的几种作出介绍。

一、Wiener 拓扑指数

最早将拓扑指数引入化学的是 Wiener[21]。他定义：分子中所有不同碳原子之间的距离总和即为 Wiener 拓扑指数，即

$$W(\text{Wiener 指数})=(\sum d_{ij})/2 \tag{9-5}$$

后来，此定义被拓广至分子中所有不同非氢原子之间的距离总和。

现仍以化合物2,3-二甲基戊烷为例来加以具体说明。可以像定义邻接矩阵那样来定义一个距离矩阵，对于2,3-二甲基戊烷，其距离矩阵 D_1 为

$$D_1=\begin{pmatrix}0&1&2&3&4&2&3\\1&0&1&2&3&1&2\\2&1&0&1&2&2&1\\3&2&1&0&1&3&2\\4&3&2&1&0&4&3\\2&1&2&3&4&0&3\\3&2&1&2&3&3&0\end{pmatrix}$$

注意到上述距离矩阵也是一个对称矩阵，所以，Wiener 指数实际就是对上述距离矩阵的上三角部分或下三角部分的所有元素求和。Wiener 指数在一定的程度上反映分子的特性，所以它首先就被用来研究饱和脂肪烃的沸点的变化规律（参见表9-10)。

二、Randic 分支指数和分子连接性指数

（一）Randic 分支指数

具有分子式为 C_6H_{14} 的五个己烷异构体的结构式、隐氢图及其对应的拓扑矩阵都不相同。

它们在结构式中的主要不同完全是由它们的分支程度不同而定，希望能找到一个“数”，把它们之间在分支程度不同的区别表示出来。Randic 指数正是这样一种简便的“分支指数”。它可由下图（图 9-2）形象地表示出来：

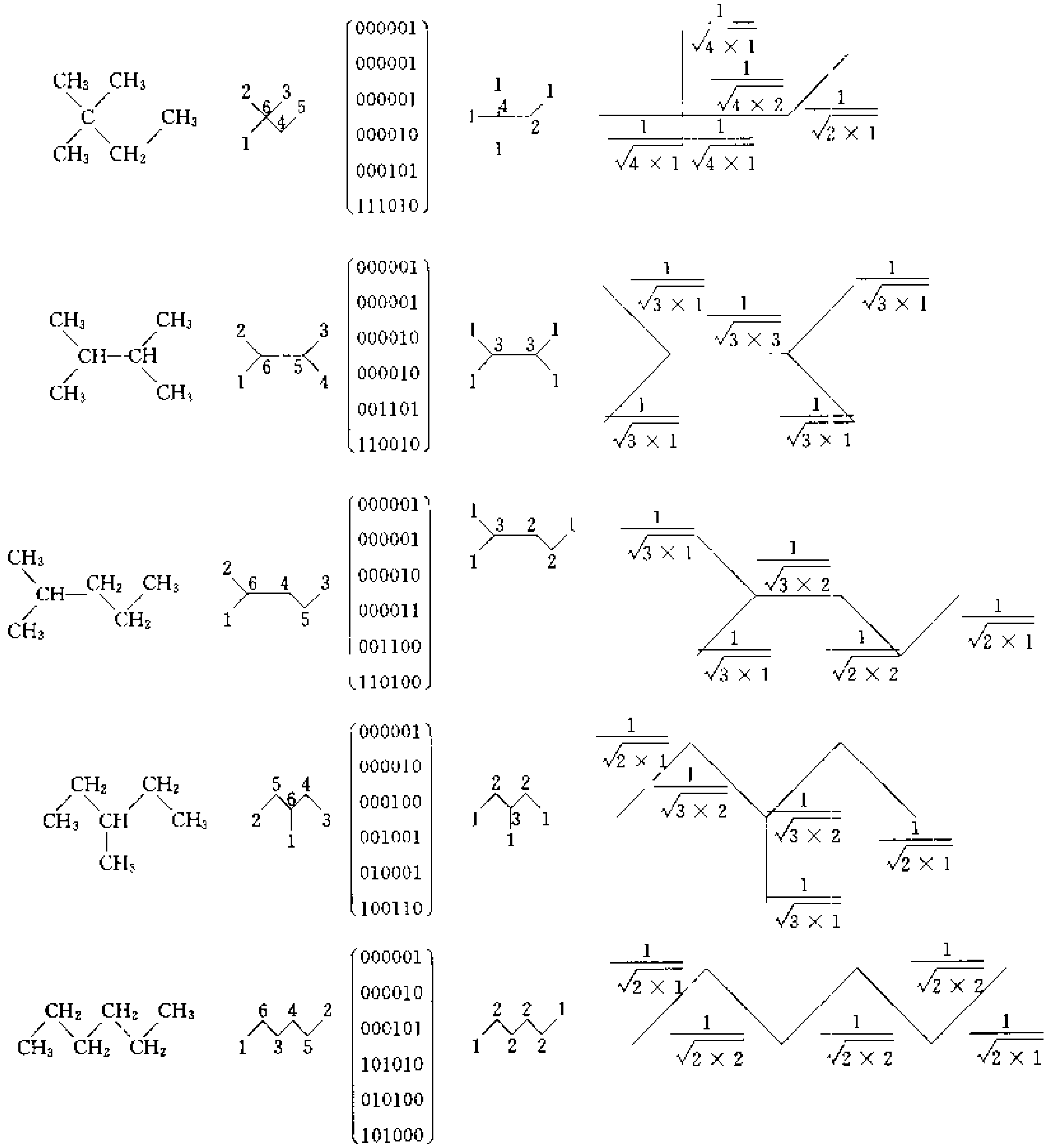


图 9-2 Randic 分支指数示意图

Randic 分支指数的定义可由下式给出：

$${}^1c=\sum(1/\sqrt{m})(1/\sqrt{n})$$

上式中的求和将包括所有的边（即键），其中 m 和 n 分别表示该边（共价键）所连的两个碳原子的点价，以符号 ${}^1c$ 表示。

${}^1c$ 可以定量反映开链饱和烃的多种物理性质，但是尚存在下列几个问题：

① 含双键或三键等化合物不能符合这个指数给定的规律；

② 环状化合物不能处理，因为键的数目比开链多；

③ 含杂原子的化合物不能包括在内。

对此，量子化学家 Kier[24]提出了分子连接性法，对 Randic 分支指数加以扩展。

（二）分子连接性指数

1．两个基本假设

Kier 提出两个基本假设：第一，表示化合物分子结构和它们若干性质间的定量关系所需要的“结构信息”包含在化合物分子的隐氢图中。对于烷烃，每个碳原子的点价 δ_i 等于 $4-h$。其中 4 是指碳原子的价电子数，h 是所连的氢原子数。杂原子的价分子连接性指数用 δ_i^* 表示，$\delta_i^*=Z^*-h$，其中 Z^* 是杂原子的价电子数，例如乙醇中的 OH，δ_{OH}^*就是 $6-1=5$。下面还给出了几个实例，其余参见表 9-5。

CH_3CH_2OH $\qquad \delta_{OH}^*=6-1=5$

$(CH_3)_3N$ $\qquad \delta_N^*=5-0=5$

$(CH_3)_3N^-$ $\qquad \delta_N^*=5-0+1=6$

表 9-5 一些杂原子的价分子连接性点价 d_i

基团	δ_i	基团	δ_i
NH_4^+	1	H_3O^+	3
NH_3	2	H_2O	4
$—NH_2$	3	—OH	5
—NH—	4	—O—	6
=NH	4	=O	6
—N—（带一支链）	5	O(硝基上氧)	6
≡N	5	O(羧酸酯、盐上氧)	6
—N—(吡啶中)	5	O(呋喃中)	6
—N—(铵中)	6	—F	(−)20
—N=(硝基中)	6	—Cl	0.690
—S—	0.944	—Br	0.254
—S—（带双键）	3.58	—I	0.085

由于卤原子具有相同的价电子数，这就产生了相同的 δ_i 值，因此根据卤原子的摩尔折射率进行修正而得到它们的经验点价 δ_i（表 9-5）。

第二个假设是：化合物分子的性质和隐氢图的连接性指数间存在某种函数关系，可用连接函数 $C_{(x)}$来反映性质（P），也就是说化合物的性质是和分子连接性有关的。可用下式表出：

$$C_{(x)}=b_0+\sum b_i(m)\,{}^m\chi_t$$

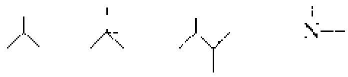
式中，$b_i(m)$可称为性质常数，是根据观察值经回归计算或理论计算而得，b_0 为一常数，${}^m\chi_t$ 为分子连接性指数，t 为子图类型，m 为图中连接的边数。

子图有下述 4 种类型：

(1) 路径项　路径项以 $t=p$ 表示，它由子图点价（只含碳原子）不大于 2 的子图构成，如包括杂原子在内，则每个点至多和两条边相连的子图叫做路径项。以下给出几个路径项的例子：



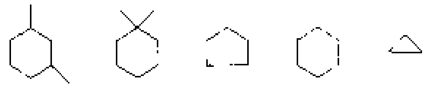
(2) 簇项　簇项以 $t=c$ 表示，它的子图点价（只含碳原子）至少为 3 或 4，但不包括点价为 2 的子图构成，假如包含杂原子，则每个中心点须和 3 条边或 4 条边相连，这样的子图叫簇项，以下给出了几个具体的例子：



(3) 路径/簇项　路径/簇项以 $t=pc$ 表示，它的子图点价除 3 或 4 外，还必须有 2 的点价存在，同样，如包含有杂原子，其定义按杂原子的路径项、簇项综合定义，具体例子如下所示：



(4) 链项　链项以 $t=\mathrm{CH}$ 表示，它的子图至少包含有一个环，下图给出了几个具体例子：



有了上述两个假设和有关定义，就可求得分子连接性指数了。

2. 分子连接性指数的求法

分子连接性指数 ${}^m\chi_t$ 是以所有不同类型的相连的子图求和而得，以下式表出：

$$ {}^m\chi_t=\sum_{i=1}^{n}{}^mS_i \tag{9-6} $$

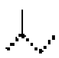
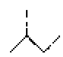
式中，n 是价为 m 的 t 类子图的数目；${}^mS_i$ 是子图项，i 是子图序号，分子拓扑性质最关键的关系是取决于点价。

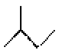
子图项计算公式如下：

$$ {}^mS_i=\Pi(\delta_j)_i^{-\frac{1}{2}} \tag{9-7} $$

从式 (9-6) 和式 (9-7) 可知，欲计算一个分子的分子连接性指数，必须剖析出所有的子图，如异戊烷的子图如表 9-6 所示。

表 9-6　异戊烷的子图表[①]

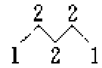
子图类型	阶 m(即边数)			
	1	2	3	4
路径项	   	   	 	
簇项				
路径/簇项				

① 它的隐氢图为 。

2. 分子连接性指数的计算举例

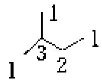
(1) 零阶项 ${}^0\chi$ 的计算　零阶项的子图是由一个点构成的子图，其边数 m 为零，计算公式如下：

$$ {}^0\chi=\sum\delta_i^{-\frac{1}{2}} $$

例如正戊烷，它的隐氢图为 

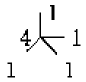
，

则 $$^{0}\chi=\sum\delta_i^{-\frac{1}{2}}=1/\sqrt{1}+1/\sqrt{2}+1/\sqrt{2}+1/\sqrt{2}+1/\sqrt{1}$$

又如异戊烷，它的隐氢图为 

，

则 $$^{0}\chi=\sum\delta_i^{-\frac{1}{2}}=1/\sqrt{1}+1/\sqrt{3}+1/\sqrt{2}+1/\sqrt{1}+1/\sqrt{1}$$

又如新戊烷，它的隐氢图为 

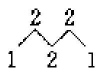
，

则 $$^{0}\chi=\sum\delta_i^{-\frac{1}{2}}=1/\sqrt{1}+1/\sqrt{1}+1/\sqrt{4}+1/\sqrt{1}+1/\sqrt{1}$$

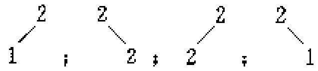
（2）一阶项$^{1}\chi$的计算　一阶项的子图是由两个点构成的子图，其边数 m 为 1，计算公式如下：

$$^{1}\chi=\sum(\delta_i\delta_j)^{-\frac{1}{2}}$$

仍举几个具体的例子来具体说明。

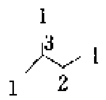
例如正戊烷，它的隐氢图为 

，

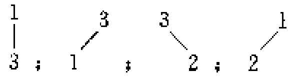
而它的子图则是 

所以，它的一阶连接性指数为

$$^{1}\chi=\sum(\delta_i\delta_j)^{-\frac{1}{2}}=1/\sqrt{1\times2}+1/\sqrt{2\times2}+1/\sqrt{2\times2}+1/\sqrt{1\times2}=2.414$$

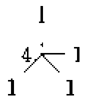
又如异戊烷，它的隐氢图为 

，

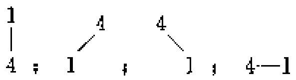
它的子图则是 

它的一阶连接性指数为

$$^{1}\chi=\sum(\delta_i\delta_j)^{-\frac{1}{2}}=1/\sqrt{1\times3}+1/\sqrt{1\times3}+1/\sqrt{3\times2}+1/\sqrt{1\times2}=2.270$$

又如新戊烷，它的隐氢图为 

，

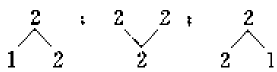
它的子图则是： 

它的一阶连接性指数为

$$^{1}\chi=\sum(\delta_i\delta_j)^{-\frac{1}{2}}=1/\sqrt{1\times4}+1/\sqrt{1\times4}+1/\sqrt{1\times4}+1/\sqrt{1\times4}=2.000$$

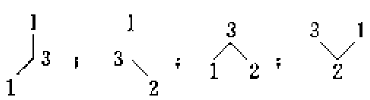
（3）二阶项$^{2}\chi$的计算　二阶项的子图是由三个点构成的子图，其边数 m 为 2，计算公式如下：

$$^{2}\chi=\sum(\delta_i\delta_j\delta_k)^{-\frac{1}{2}}$$

例如正戊烷，它的子图是 

所以，它的二阶连接性指数为

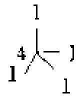
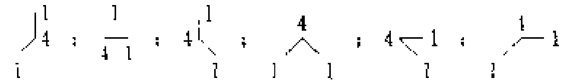
$$^{2}\chi=\sum(\delta_i\delta_j\delta_k)^{-\frac{1}{2}}=1/\sqrt{1\times2\times2}+1/\sqrt{2\times2\times2}+1/\sqrt{2\times2\times1}=1.354$$

对于异戊烷，它的子图是 

。

二阶连接性指数为

$$^{2}\chi=\sum(\delta_i\delta_j\delta_k)^{-\frac{1}{2}}=1/\sqrt{1\times3\times1}+1/\sqrt{1\times3\times2}+1/\sqrt{1\times3\times2}+1/\sqrt{3\times2\times1}=1.802$$

对于新戊烷，它的隐氢图为 ，它的子图则是 

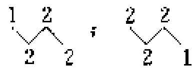
二阶连接性指数为

$$^{2}\chi=\sum(\delta_i\delta_j\delta_k)^{-\frac{1}{2}}=6\times(1/\sqrt{1\times4\times1})=3.000$$

（4）三阶项$^{3}\chi$的计算　三阶项的子图一般是由四个点构成的子图，其边数 m 为 3，计算公式如下：

$$^{3}\chi=\sum(\delta_i\delta_j\delta_k\delta_l)^{-\frac{1}{2}}$$

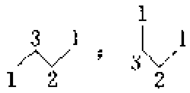
三阶项可能有 3 种类型的子图，即路径项、簇项和链项。如果是链项，则三阶项是由三个点、三条边所组成。

下面以正戊烷，异戊烷和新戊烷为例来说明三阶项的计算。对于正戊烷，它只有路径项，其子图是 

所以，它的三阶连接性指数为

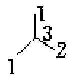
$$^{3}\chi=\sum(\delta_i\delta_j\delta_k\delta_l)^{-\frac{1}{2}}=1/\sqrt{1\times2\times2\times2}+1/\sqrt{2\times2\times2\times1}=0.707$$

但对于异戊烷，它的三阶项有两种，一为路径项，其子图有



故其三阶路径项为

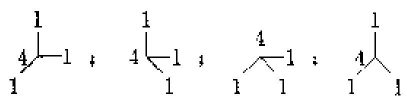
$$^{3}\chi_p=\sum(\delta_i\delta_j\delta_k\delta_l)^{-\frac{1}{2}}=1/\sqrt{1\times3\times2\times1}+1/\sqrt{1\times3\times2\times1}=0.816$$

它的簇项子图为 

故其三阶簇项为

$$^{3}\chi_c=\sum(\delta_i\delta_j\delta_k\delta_l)^{-\frac{1}{2}}=1/\sqrt{1\times3\times2\times1}=0.408$$

对于新戊烷，则只有三阶簇项，而无三阶路径项。其三阶簇项子图为



故其三阶簇项为

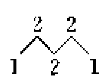
$$^{3}\chi_c=\sum(\delta_i\delta_j\delta_k\delta_l)^{-\frac{1}{2}}=4\times(1/\sqrt{1\times1\times1\times4})=2.000$$

（5）四阶项$^{4}\chi$的计算：四阶项的子图一般是由五个点构成的子图，其边数 m 为 4，计算公式如下：

$$^{4}\chi=\sum(\delta_i\delta_j\delta_k\delta_l\delta_q)^{-\frac{1}{2}}$$

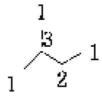
四阶项可能有 4 种类型的子图，即路径项、簇项、链项及路径/簇项。如果是链项，则四阶项是由 4 个点、4 条边所组成。

对于正戊烷，它只有路径项，其子图即为隐氢图



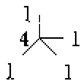
所以，它的四阶连接性指数为

$$^4\chi_p=\sum(\delta_i\delta_j\delta_k\delta_l\delta_q)^{-\frac{1}{2}}=1/\sqrt{1\times2\times2\times2\times1}=0.354$$

异戊烷则既无四阶路径项$^4\chi_p$又无四阶簇项$^4\chi_c$，而只有四阶路径/簇项$^4\chi_{pc}$，其四阶簇项/链项子图也就是隐氢图 

它的四阶连接性指数为

$$^4\chi_{pc}=\sum(\delta_i\delta_j\delta_k\delta_l\delta_q)^{-\frac{1}{2}}=1/\sqrt{1\times1\times3\times2\times1}=0.408$$

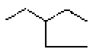
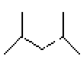
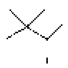
新戊烷无四阶路径项$^4\chi_p$又无四阶路径/簇项$^4\chi_{cp}$，而只有四阶簇项$^4\chi$，其四阶簇项$^4\chi_c$子图也是隐氢图 

它的四阶连接性指数为

$$^4\chi_c=\sum(\delta_i\delta_j\delta_k\delta_l\delta_q)^{1/2}=1/\sqrt{1\times1\times1\times1\times4}=0.509$$

表 9-7 列出了庚烷各种异构体的各类子图数目。

表 9-7　庚烷异构体路径、簇、路径/簇项的子图数目表

隐氢图	$^1\chi_p$	$^2\chi_p$	$^3\chi$		$^4\chi$			$^5\chi$			$^6\chi$			总		
			p	c	p	c	pc	p	c	pc	p	c	pc	p	c	pc
	6	5	4	0	3	0	0	2	0	0	1	0	0	21	0	0
	6	6	6	1	3	0	3	0	0	2	0	0	1	21	1	6
	6	6	5	1	3	0	2	1	0	2	0	0	0	21	1	4
	6	6	4	1	3	0	1	2	0	1	0	0	1	21	1	3
	6	7	6	2	2	0	5	0	1	3	0	0	1	21	3	9
	6	7	4	2	4	0	2	0	0	4	0	0	1	21	2	7
	6	8	6	4	1	1	6	0	0	4	0	0	1	21	5	11
	6	8	4	4	3	1	3	0	0	4	0	0	1	21	5	8
	6	9	6	5	0	1	9	0	3	2	0	1	0	21	10	11

三、分子识别数

无论是分子的 Wiener 指数还是分子的 Randic 的分支指数都存在着退化性或称简并性，即不同的分子具有相同的指数值，如在表 9-10 中，2，2-二甲基戊烷的 Wiener 指数为 46，而 2，3-二甲基戊烷的 Wiener 值数也为 46，这就说明在 Wiener 指数指标中存在着退化性，像这样的例子还可举出不少；Randic 的分支指数的退化性没有 Wiener 指数那么严重，但也存在着同样的退化性，3-甲基庚烷与 4-甲基庚烷就可产生同样的 Randic 分支指数。所以，在 1984 年，

Randic 又提出了分子识别数 (MIDN) 这一新的分子拓扑指数[25]。分子识别数这一指标实际上是分子连接性指数和分子路径数的一种结合，或更精确的说，是一种加权的分子路径数，其中权值就与分子连接性相关[26]。在给出分子识别数的定义之前，先看看有关分子和原子的路径数。表 9-8 给出了 2-甲基了烷的全部有关路径数。有关该表的具体理解可参看图 9-3。结合表 9-8 和图 9-3，可以看到，在计算路径数时，实际上将只有一个点，即路径等于零时，也算存在一个路径数；另外，在计算路径大于零时的路径数时，分子的路径数是原子路径数的二分之一，这是因为在计算原子路径数时，同一路径算了两次（参见图 9-3)。在介绍了路径数概念基础上，分子识别数的数学定义可由下式给出：

$$\mathrm{MIDN}=\sum_{j} w_{0j}+\frac{1}{2}\sum_{k} w_{ik} \tag{9-8}$$

式中，w 是路径权重；下标 0j 表示对子所有长度为零的路径，而下标 ik 则表示所有长度大于零的路径。

表 9-8　2-甲基丁烷的原子和分子路径数一览表

原子编号	路径长度数					总　和
	0	1	2	3	4	
1	1	1	2	1	0	5
2	1	3	1	0	0	5
3	1	2	2	0	0	5
4	1	1	1	2	0	5
5	1	1	2	1	0	5
分子路径数	5	4	4	2	0	
分子路径数总和：15						

有关权重的计算由下述原则确定：

① 长度为零的路径的权重都为 1；

② 长度大于零的路径的权重由其分子连接性指数确定。

表 9-9 给出了 2-甲基了烷的全部有关路径权重。从表 9-9 中可以看出，2-甲基丁烷的分子识别数为 8.6968019。

表 9-9　2-甲基丁烷的原子和分子路径权重一览表

原子编号	路径长度数					总　和
	0	1	2	3	4	
1	1	0.5774	0.5690	0.1667	0	2.3131
2	1	1.5629	0.2887	0	0	2.8516
3	1	1.1154	0.4714	0	0	2.5868
4	1	0.7071	0.2887	0.3333	0	2.3291
5	1	0.5774	0.5690	0.1667	0	2.3131
分子路径权重	5	2.2701	1.0934	0.3333	0	
分子识别数：8.6968019						

表 9-10 列出了部分 C_4～C_7 烷烃的沸点、Wiener 指数、Randic 分支指数和分子识别数，从表 9-10 可以看出，烷烃的沸点与 Wiener 指数、Randic 分支指数和分子识别数都存在较好的线性相关性，其中沸点与 Wiener 指数的相关系数为 0.971；沸点与 Randic 分支指数的相关系数为 0.994；分子识别数与沸点的相关性最好。

原子序号	路径长度			
	0	1	2	3
1	1	1—2	1—2—3 1—2—5	1—2—3—4
2	2	2—1 2—3 2—5	2—3—4	
3	3	3—2 3—4	3—2—5 3—2—1	
4	4	4—3	4—3—2	4—3—2—1 4—3—2—5
5	5	5—2	5—2—1 5—2—3	

2-甲基丁烷的隐氢图

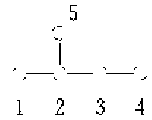


图 9-3 与表 9-8 相对应的路径子图

表 9-10 部分 C_4～C_7 烷烃的沸点、Wiener 指数、Randic 分支指数和分子识别数

化合物	沸点	Wiener 指数	Randic 分支指数	分子识别数
2-甲基丙烷	−11.73	9	1.732	6.7321
正丁烷	−0.50	10	1.914	6.8713
2,2-二甲基丙烷	9.50	16	2.000	8.5000
2-甲基丁烷	27.85	18	2.270	8.6968
正戊烷	36.07	20	2.414	8.8499
2,2-二甲基丁烷	49.74	28	2.561	10.4660
2,3-二甲基丁烷	57.99	29	2.643	10.5236
2-甲基戊烷	60.27	32	2.770	10.6792
3-甲基戊烷	63.28	31	2.808	10.6759
正己烷	68.74	35	2.914	10.8392
2,2-二甲基戊烷	79.20	46	3.061	12.4490
2,4-二甲基戊烷	80.50	48	3.126	12.5092
2,2,3-三甲基丁烷	80.88	42	2.943	12.2931
3,3-二甲基戊烷	86.03	44	3.121	12.4427
2,3-二甲基戊烷	89.78	46	3.181	12.5052
2-甲基己烷	90.05	52	3.270	12.6704
3-甲基己烷	91.85	50	3.308	12.6600
3-乙基戊烷	93.48	48	3.346	12.6692
正庚烷	98.42	56	3.414	12.8338

为计算分子识别数，Randic，Wilkins 等提供了一个计算该值的 Basic 程序，用它来计算一般分子的分子识别数将比较容易[27]。

Randic 对 400 多个分子结构，其中包括 10 个碳原子以下的所有烷烃进行了分子识别数的计算，没有发现退化性。1985 年，Trinajstic 等对包括 20 个碳原子以下的 618 000 种烷烃结构进行了系统地计算，发现其中有 124 种化合物的两个异构体和 1 种化合物的 3 个异构体分别具有相同的分子识别数，如 2,3-二甲基-6-乙基-5-异丙基辛烯与 2,6-二甲基-5-乙基-3-异丙基辛烯就是其中的一个例子[28]，它们具有完全相同的分子识别数。Randic 后又对分子识别数进行了改进，提出了第二分子识别数，读者可参阅文献[29]。

四、苏尔兹分子拓扑指数

1989年，苏尔兹（Schultz）又提出了称为分子拓扑指数（MTI）的新指标[30]，这一拓扑指数的优点是它的退化性不严重，很多上述方法中具有相同特征数的分子对，它们的Schultz分子拓扑指数是不相同的。为方便起见，现仍以2-甲基丁烷为例来说明Schultz分子拓扑指数构成方法。2-甲基丁烷的隐氢图示于图9-3，它的距离矩阵（$\boldsymbol{D}$）、邻接矩阵（$\boldsymbol{A}$）和各原子的点价矢量（$\boldsymbol{v}$）如下：

$$\boldsymbol{D}=\begin{pmatrix}0&1&2&3&2\\1&0&1&2&1\\2&1&0&1&2\\3&2&1&0&3\\2&1&2&3&0\end{pmatrix}\qquad \boldsymbol{A}=\begin{pmatrix}0&1&0&0&0\\1&0&1&0&1\\0&1&0&1&0\\0&0&1&0&0\\0&1&0&0&0\end{pmatrix}$$

$$\boldsymbol{v}=[1\quad 3\quad 2\quad 1\quad 1]$$

Muller等曾指出[31]，对邻接矩阵的每一行或者是每一列加合即可得点价矢量$\boldsymbol{v}$。在此，Schultz是用的$\boldsymbol{D}$矩阵和$\boldsymbol{A}$矩阵的和，即（$\boldsymbol{D}+\boldsymbol{A}$）

$$\boldsymbol{D}+\boldsymbol{A}=\begin{pmatrix}0&1&2&3&2\\1&0&1&2&1\\2&1&0&1&2\\3&2&1&0&3\\2&1&2&3&0\end{pmatrix}+\begin{pmatrix}0&1&0&0&0\\1&0&1&0&1\\0&1&0&1&0\\0&0&1&0&0\\0&1&0&0&0\end{pmatrix}=\begin{pmatrix}0&2&2&3&2\\2&0&2&2&2\\2&2&0&2&2\\3&2&2&0&3\\2&2&2&3&0\end{pmatrix}$$

继续对（$\boldsymbol{D}+\boldsymbol{A}$）与$\boldsymbol{v}$的乘积所得矢量的所有元素求和就得到了Schultz分子拓扑指数：

$$\boldsymbol{v}(\boldsymbol{D}+\boldsymbol{A})=[15\quad 10\quad 12\quad 16\quad 15]$$

而

$$\mathrm{MTI}=15+10+12+16+15=68 \tag{9-9}$$

Schultz对42种不同的烷烃进行了分子拓扑指数的计算，所得结果示于表9-11。在表9-11中的第17号和第18号，第19号和第20号、第24号和第25号、第27号和第28号、第29号和第31号、第31号和第32号、第21号和33号、第41号和第42号这些分子对都是别的分子描述指数相同而难区分的分子对。虽然，Schultz提出的分子拓扑指数在分子结构描述的退化性有所进步，但是，对于此分子拓扑指数与沸点的相关性却不如Randic分支指数。为此，Schultz提出使用($\boldsymbol{D}+\boldsymbol{A}$)矩阵的行列式值和第一特征值等数量指标[32~34]来描述分子特性，以改善它们与分子化学性质的相关性。在此将作进一步介绍。

仍以2-甲基丁烷为例来进行说明。对于2-甲基丁烷的距离矩阵$\boldsymbol{D}$和（$\boldsymbol{D}+\boldsymbol{A}$）矩阵，它们的行列式分别为32和176；它们的特征值分别为：$\lambda_1=7.45929$；$\lambda_2=-3.86269$；$\lambda_3=-2.0000$；$\lambda_4=-1.08462$；$\lambda_5=-0.51198$（$\boldsymbol{D}$矩阵）；$\lambda_1=8.85345$；$\lambda_2=-3.38547$；$\lambda_3=-2.0000$；$\lambda_4=-2.0000$；$\lambda_5=-1.46798$（$\boldsymbol{D}+\boldsymbol{A}$矩阵）。在此没有提到拓扑邻接矩阵$\boldsymbol{A}$，这是因为Trinajstic已

对此进行过研究，他所得结果说明拓扑连接矩阵 A 的特征值是一个退化性很高的指标，很多不同化合物的拓扑邻接矩阵的特征值相同[35]。

表 9-11 部分 C_1～C_{15}烷烃的沸点和分子拓扑指数

编号	化合物	沸点	分子拓扑指数	编号	化合物	沸点	分子拓扑指数
1	甲烷	−164	0	22	2,2,3-三甲基丁烷	80.88	156
2	乙烷	−88.6	4	23	正辛烷	125.7	306
3	丙烷	−42	16	24	3-甲基庚烷	119	276
4	正丁烷	−0.50	38	25	4-甲基庚烷	117.7	272
5	2-甲基丙烷	−11.73	36	26	2,5-二甲基己烷	109.1	270
6	正戊烷	36.07	74	27	2,2-二甲基己烷	106.9	260
7	2-甲基丁烷	27.85	68	28	2,4-二甲基己烷	109.5	258
8	2,2-二甲基丙烷	9.50	64	29	3,4-二甲基己烷	117.7	246
9	正己烷	68.74	128	30	3,3-二甲基己烷	112	244
10	2-甲基戊烷	60.27	118	31	3-乙基-2-甲基戊烷	115.7	242
11	3-甲基戊烷	63.28	114	32	3-乙基-3-甲基戊烷	118.3	232
12	2,3-二甲基丁烷	57.99	108	33	2,2,3,3-四甲基丁烷	106.3	214
13	2,2-二甲基丁烷	49.74	106	34	壬烷	150.8	438
14	正庚烷	98.42	204	35	癸烷	174.1	604
15	2-甲基己烷	90.05	190	36	十一烷	198.8	808
16	3-甲基己烷	91.85	182	37	十二烷	216.3	1054
17	2,4-二甲基戊烷	80.50	176	38	十三烷	235.4	1346
18	3-乙基戊烷	93.48	174	39	十四烷	253.7	1688
19	2,2-二甲基戊烷	79.20	170	40	十五烷	270.6	2084
20	2,3-二甲基戊烷	89.78	168	41	2,6-二甲基-5-乙基-3-异丙基辛烷		1366
21	3,3-二甲基戊烷	86.03	162	42	2,3-二甲基-6-乙基-5-异丙基辛烷		1358

表 9-12 列出了对前述 42 种不同的烷烃的距离矩阵 D 和（$D+A$）矩阵的行列式和它们相应的第一特征值及用它们分别计算所得的沸点值。为列表方便在此略去了这些化合物的名称，只保留了它们的编号。由（$D+A$）矩阵的行列式绝对值的 lg 值的平方根对相应沸点值进行回归计算的相关系数为 0.994；而由距离矩阵 D 的第一特征值之 lg 值对相应沸点值进行回归计算的相关系数为 0.969；由（$D+A$）矩阵的第一特征值之 lg 值对相应沸点值进行回归计算的相关系数为 0.987，说明采用这些数值来计算可提高 Schultz 分子拓扑指数与相应化学性质的相关性。

表 9-12 部分 C_1～C_{15}烷烃的沸点和距离矩阵 D 及矩阵 $|D+A|$ 特征值之间的关系

编号	沸点	$\|D\|$	$\|D+A\|$	由 $\|D+A\|$ 计算的沸点值	λ_1		由 $\lambda_1(D+A)$ 算的沸点值
					$D\|$	$\|D+A\|$	
1	−161	0	0		0	0	
2	−88.6	−1	−4	−108.8	1	2	−119.3
3	−42	4	6	−47.4	2.7321	4	−48.6
4	−0.50	−12	−60	−2.3	5.1623	6.53113	1.4
5	−11.73	−12	−48	−9.4	4.6458	6	−7.3
6	36.07	32	224	35.8	8.2882	9.72970	42.0
7	27.85	32	176	29.2	7.4593	8.85345	32.4
8	9.50	32	128	20.2	6.6056	8	22.1
9	68.74	−80	−817	68.9	12.109	13.6116	76.3
10	60.27	−80	−656	63.5	11.059	12.5154	67.7

续表

编号	沸点	$\|D\|$	$\|D+A\|$	由 $\|D+A\|$ 计算的沸点值	λ_1		由 $\lambda_1(D+A)$ 算的沸点值
					$\|D\|$	$\|D+A\|$	
11	63.28	−80	−644	63.1	10.712	12.1719	64.9
12	57.99	−80	−512	57.3	1[illegible]	11.4031	58.2
13	49.74	−80	−461	54.8	9.6702	11.0769	55.3
14	98.42	192	2840	97.9	16.625	18.1783	105.8
15	90.05	192	2372	93.8	15.405	16.9205	98.5
16	93.5	192	2352	93.6	14.297	15.7492	91.2
17	92	192	2348	93.6	14.864	16.3497	95.0
18	80.5	192	1920	88.4	14.176	15.6472	90.5
19	79.20	192	1728	86.6	13.635	15.0970	86.9
20	89.78	192	1872	88.4	13.634	15.0731	86.7
21	86.03	192	1680	86.0	13.070	14.4942	82.7
22	80.88	192	1344	80.8	12.394	13.8023	77.7
23	125.7	−448	−9486	123.8	21.836	23.4316	131.7
24	119	−448	−8127	120.6	19.763	21.3026	122.0
25	117.7	−448	−8160	120.7	19.542	21.0738	120.9
26	109.1	−448	−6864	117.0	19.112	20.6428	118.8
27	106.9	448	−6220	115.0	18.413	19.9327	115.2
28	109.5	−448	−6816	116.9	18.396	19.8985	115
29	17.7	−448	−6700	116.5	17.676	19.1471	111.1
30	12	−448	−6124	114.6	17.443	18.9193	109.9
31	15.7	−448	−6832	116.9	17.419	18.8774	109.6
32	118.3	−448	−6076	114.5	16.671	18.1096	105.4
33	106.3	−448	−3520	102.6	14.937	16.3459	95.0
34	150.8	1024	31007	147.7	27.742	29.3729	154.7
35	174.1	−2304	−100095	170.0	34.343	36.0037	175.5
36	196.8	5120	320048	191.1	41.638	43.3249	194.4
37	216.3	−11264	−1014203	211.1	49.629	51.3375	211.7
38	235.4	24576	3188230	230.0	58.314	60.0420	227.7
39	253.7	−53248	−9954933	248.2	67.694	69.4390	242.5
40	270.6	114688	30910288	265.6	77.769	79.5288	256.3
41		114688	17734496	257.1	51.924	53.5129	215.9
42		114687	16622076	256.1	52.136	53.7266	216.3

最近，Klavzar 和 Gutman 对 Schultz 分子拓扑指数与 Wiener 指数的关系进行了研究[36]，他们证明了 Schultz 的分子拓扑指数与 Wiener 指数存在如下关系：

$$\alpha W \leqslant \mathrm{MTI} \leqslant \beta W$$

具体地说，可由以下两个表达式具体给出：

① 如果 G 为一连接图，则

$$2v_{\min}W(\mathrm{G}) < \mathrm{MTI}(\mathrm{G}) \leqslant 2v_{\max}W(\mathrm{G})$$

式中，$v_{\min}$ 和 $v_{\max}$ 分别表示最小和最大的点价数。

② 如果分子结构隐氢图 G 为一多环芳烃体系，则有

$$4W(\mathrm{G}) < \mathrm{MTI}(\mathrm{G}) \leqslant lW(\mathrm{G})$$

式中，$l=6+6/(\sqrt{15\times25})=6.9295$。

① ② 两个关系表明，Schultz 分子拓扑指数与 Wiener 指数之间存在一定的线性相关关系。事实上，对于烷烃来说，在 Schultz 分子拓扑指数与 Wiener 指数之间存在一定的线性相关

关系已经有人进行过系统的研究[37]。

巴纳班曾在一篇评论文章中指出，用实数来表达顶点不变数的确应可看成是第三代的分子拓扑指数。据他的评论称[24]，第一代分子拓扑指数是以对局部顶点不变数（LOVT's）的简单操作而得的整数；第二代分子拓扑指数是对由那些对局部顶点不变数的整数进一步进行复杂操作而得到的实数；第三代分子拓扑指数则应是那些从第二代分子拓扑指数进一步操作而得到的实数。关于此三类分子拓扑指数的具体方法及其特性已列于表 9-13 之中。

表 9-13　基于局部顶点不变数（LOVT's）的第三代分子拓扑指数

分子拓扑指数代	LOVI	操　作	分子拓扑指数形式	退化性或简并性
第一代	整数	简单标记	整数	高
第二代	整数或有理数	结构性计算	实数	中等或低
第三代	实数	两形式皆有	实数	低

按照巴纳班的分类方法，像 Wiener 指数就应为第一代分子拓扑指数，第一代分子拓扑指数的例子还有：

① Zagreb 的基团拓扑指数，记为 M[38]

$$M = \sum_i \delta_i^2 \tag{9-10}$$

② Gordon-Scantlebury 指数，记为 N[39]

$$N = \sum_i \delta_i(\delta_i - 1)/2 \tag{9-11}$$

③ 平方指数，记为 Q[40]

$$Q = [\sum_i (i^2 - 2i)\delta_i + 2]/2 \tag{9-12}$$

在(9-10)～(9-12)三式中，δ_i 均表示为分子图的点价。实际上，(9-10)～(9-12)三式存在着以下的线性关系：

$$Q = N - n + 2 = 3 - 2n + (M/2)$$
$$M = 2(N + n - 1)$$

式中，n 为分子隐氢图中的顶点数。

一般说来，第一代分子拓扑指数都具有较高的退化性。

第二代分子拓扑指数的典型代表是 Randic 的分支指数和后由 Kier 拓广的分子连接性指数，另外一个例子是分子图的所有顶点的均方距离指标，记为 $D^{(2)}$[41]：

$$D^{(2)} = (\sum_{i,j}\delta_{ij}^2)^{1/2}/[n(n-1)] \tag{9-13}$$

式中，δ_{ij}表示顶点之间的距离，n 表示分子图中顶点的数目。

第三代分子拓扑指数可以是这样一些指标，它们的局部顶点不变数是由实数表示的，这样的例子可以是邻接矩阵或距离矩阵的特征值、行列式值等。在此仅介绍一种由 Balaban 等提出的这样的第三代分子拓扑指数[23]，他们称之为回归距离和及回归顶点点价。

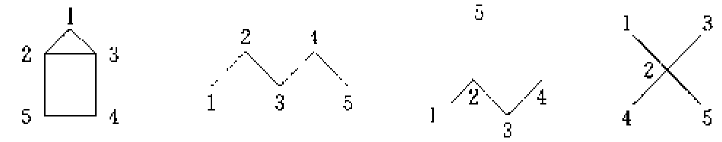
五、回归距离和回归顶点点价

因顶点点价反映了该顶点与其相连顶点的关系；而顶点之间的距离和则反映了一些更远距离顶点对该顶点的影响，所以，分子拓扑图中的顶点点价和顶点之间的距离和已在第一代和第二代的分子拓扑指数中得到了相当广泛地应用。巴纳班等在这里则通过回归顶点点价来

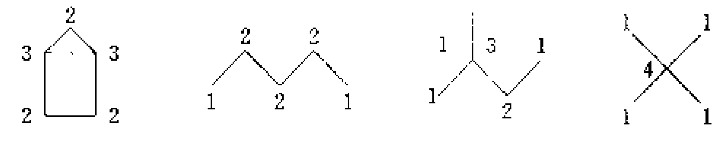
增强顶点点价的作用，使它能将一些更远的顶点的贡献也被包括进来，当然，随着距离的增大，其作用也将相应减少；而回归距离和则是起到相应的增强简单顶点和的作用，而且，经此增强后，这些指标都成了实数，这样就可以达到减少拓扑指标退化性的目的。下面将通过两个简单的例子来加以说明。

图 9-4 和图 9-5 分别给出了几个具体构成回归顶点点价和回归距离和指标的例子。

顶点编号



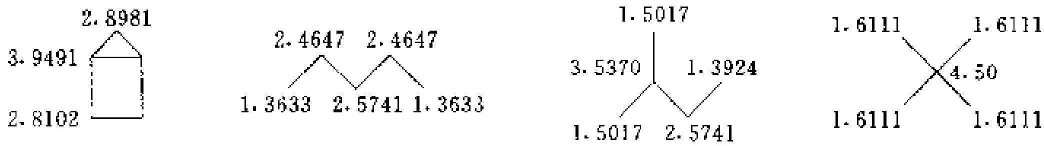
顶点点价



矩阵

j \ i	1	2	3	1	2	3	4	5	1	2	3	4	1	2	3
1	2	6	4	1	2	2	2	1	1	3	3	1	1	4	3
2	3	7	2	2	3	2	1	0	3	1	1	0	4	4	0
3	3	7	2	2	1	2	0	0	2	4	2	0	1	4	3
4	2	5	5	2	3	2	1	0	1	2	3	2	1	4	3
5	2	5	5	1	2	2	2	1	1	3	3	1	1	4	3

$RVD_j^{(1)}$



$RVD_j^{(2)}$

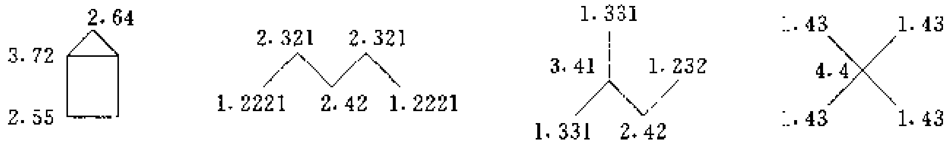


图 9-4　回归顶点点价计算示意图

从图 9-4 和图 9-5 可以看出，为四张具有五个碳原子的隐氢图进行了编号，并将这些碳原子的点价或各顶点到该点的距离和标出。然后，构成了一个矩阵，在该矩阵的第一列中，简单地计下该顶点相应的点价和距离，而在该矩阵的第二列中，则将与其最邻近的顶点所标出的点价或距离进行加合，即将只与该顶点直接相连（距离等于 1 的）的顶点的相应数加合；对于该矩阵的第 i 列，则将与该顶点以（$i-1$）个边相连（距离等于（$i-1$）的）的顶点的相应数加合。两个顶点之间的最大距离称为该分子图的直径，它也等于该矩阵的列数。

得到这两个矩阵后，相应的 RVD_j 和 RDS_j 则可通过以下两式来进行计算：

$$LOVI_j^{(1)}=\sum i^{-3}y_{ij} \tag{9-14}$$

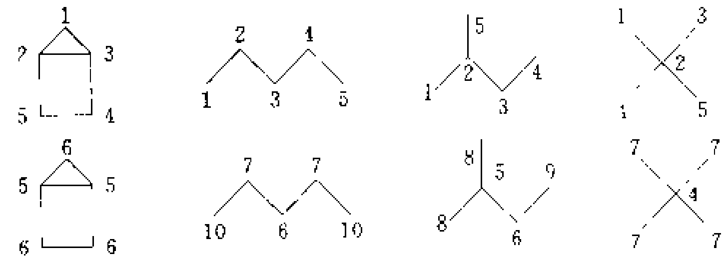
$$LOVI_j^{(2)}=\sum 10^{1-i}y_{ij} \tag{9-15}$$

在此，y_{ij}表示前述矩阵中的第 i 行第 j 列的元素。由（9-14）（9-15）式计算出来的局部顶点不变数在图 9-4 和图 9-5 的最后两行给出。这样所得到的分子拓扑指数将具有很小的退化性。用这些指标计算了 35 个烷烃（$n=5-8$）的沸点，所得的相关系数为 0.985，可见它们也具有很

好的化学特性表达功能。

顶点编号

距离和



j \ i	1	2	3	1	2	3	4	5	1	2	3	4	1	2	3
1	6	10	12	10	7	6	7	10	8	5	14	9	7	4	21
2	5	17	6	7	16	7	10	0	5	22	9	0	4	28	0
3	5	17	6	6	14	20	0	0	6	11	16	0	7	4	21
4	6	11	11	7	16	7	10	0	9	6	5	16	7	4	21
5	6	11	11	10	7	6	7	10	8	5	14	9	7	4	21

矩阵

$RDS_j^{(1)}$

$RDS_j^{(2)}$

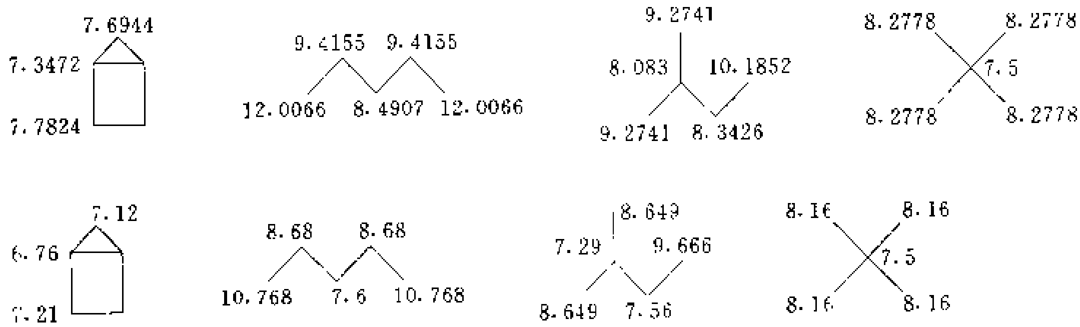


图 9-5　回归距离和计算示意图

第三节　构效关系建模方法

一、基于回归分析的建模方法

由前述方法得到了分子结构的数量描述指标后，一个最直接的方法就是用回归方程将这些分子结构的数量描述指标与化合物的某种性质联系起来，并用最小二乘的方法得到相应的回归参数（α_1，α_2，…，α_n），即

$$y=\alpha_1x_1+\alpha_2x_2+\cdots\alpha_nx_n \tag{9-16}$$

可以说目前很多构效定量关系的研究都基于此。下面将摘要地介绍这方面的一些具体应用。

（1）分子连接性指数与立体参数间的关系　由 Taft 提出的取代基立体效应参数 E_s 值是一个经验值，不少基团的 E_s 值用类比法还难以得到，所以使用受到严重限制。对已知 E_s 值的 18 个脂肪酯类烃基部分，用分子连接性指数可与 E_s 值建立如下相关关系方程式[42]：

$$E_s=-0.544\,{}^2\chi-1.40\,{}^3\chi+1.09\,{}^4\chi+0.403$$

其复相关系数 $R=0.961$。从逻辑上来看，反应中心（如酯基）周围的基团由于立体效应而影响反应速度，因此用${}^2\chi$、${}^3\chi$ 等延伸项可反映这种聚集现象，即反映与反应中心相连的基团的多

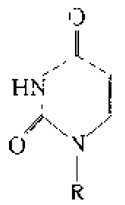
少及其立体效应的大小。

（2）分子连接性指数与胸腺嘧啶磷酸酯酶抑制剂活性关系　11 个 N_1 位烃基和芳烃基取代的胸腺嘧啶衍生物（参见表 9-14）的活性与 $^1\chi$ 的 QSAR 方程式如下：

$$\lg\left(\frac{1}{c}\right)=0.366\,^1\chi-3.364$$

其复相关系数为 0.920 式中 c 为物质的浓度（$mol \cdot L^{-1}$）。

表 9-14　胸腺嘧啶磷酸酯酶抑制剂 $^1\chi$ 值和活性值

	R	$^1\chi$	$\lg\frac{1}{c}$	
			观测值	计算值
	甲基	3.698	−2.30	−2.01
	丁基	5.236	−1.35	−1.44
	异戊基	5.592	−1.30	−1.31
	环戊基	5.270	−1.28	−1.43
	异己基	6.092	−1.17	−1.13
	戊基	5.736	−1.15	−1.26
	3-苯丙基	7.254	−1.11	−0.71
	2-苯丙基	6.754	−0.80	−0.89
	苯甲基	6.254	−0.76	−1.07
	4-苯丁基	7.754	−0.60	−0.52
	5-苯戊基	8.254	−0.32	−0.34

（3）分子连接性指数与 2-咪唑烷酮类衍生物的活性关系　2-咪唑烷酮类衍生物为中枢神经系统抑制剂。相秉仁等[43]用分子连接性指数与 $\lg\frac{1}{c}$ 对 21 个化合物建立的 QSAR 方程式，获得显著效果。

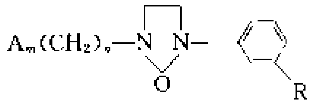
$$\lg\frac{1}{c}=0.2455\,^1\chi-6.3933\,^5\chi_{PCT}-13.1189\,^6\chi_{CYT}+3.4448$$

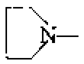
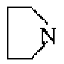
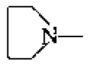
式中，$^5\chi_{PCT}$为咪唑烷酮 2 位接有二甲胺边链的化合物采用的末端路径/簇项值；$^6\chi_{CYT}$为含有哌啶环和吗啉环边链的化合物的末端 6 原子回路项。应用分子连接性指数计算所得结果见表 9-15。

从表 9-15 可以看出，分子中含有二甲胺基侧链可使活性降低（氯丙嗪例外，因母核结构不同，不宜类推）。这从 QSAR 方程式中可以定量反映出来，路径/簇项值$^5\chi_{PCT}$的系数是负值，表示它使母体活性降低，一般要降低 0.64 对数单位值（化合物 9～15）和 1.01 对数单位值（化合物 16 与 17）。

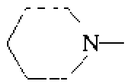
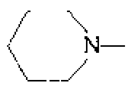
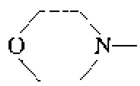
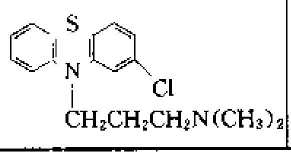
含有六元哌啶环与吗啉环的也使活性降低，方程式中$^6\chi_{CYT}$项的系数亦为负数，提示这两种六元环基团对活性呈负变影响。一般要降低 1.03 对数单位值（化合物 18 与 19）和 0.59 对数单位值（化合物 20）。这可能是六元环立体障碍较大而妨碍分子与受体的嵌合。

表 9-15　分子连接性指数与咪唑烷酮类衍生物的中枢抑制活性 QSAR 方程式的计算结果



编号	A_m	n	R	$^{1}\chi$	$^{5}\chi_{PCT}$	$^{6}\chi_{CY}$	$\lg\left(\frac{1}{c}\right)$ 实验	$\lg\frac{1}{c}$ 计算	$\left\lvert\Delta\lg\frac{1}{c}\right\rvert$
1		2	Br	8.0864	0.0000	0.0000	5.63	5.43	0.20
2		2	Cl	7.6958	0.0000	0.0000	5.47	5.33	0.14
3		2	H	7.1788	0.0000	0.0000	5.01	5.20	0.19
4	$(CH_3CH_2)(CH_3CH_2)N$—	2	Br	7.9993	0.0000	0.0000	5.42	5.41	0.01
5	$(CH_3CH_2)(CH_3CH_2)N$—	2	Cl	7.6081	0.0000	0.0000	5.33	5.31	0.02
6	$(CH_3CH_2)(CH_3)N$—	2	C	7.0345	0.0000	0.0000	5.37	5.19	0.18
7	$(CH_3CH_2)(CH_3)N$—	2	H	6.5208	0.0000	0.0000	5.09	0.05	0.04
8	$(CH_3CH_2)(CH_3)N$—	2	Br	7.4222	0.0000	0.0000	5.13	5.27	0.14
9	$(CH_3)(CH_3)N$—	2	Cl	6.4671	0.1000	0.0000	5.03	4.39	0.64
10	$(CH_3)(CH_3)N$—	2	Br	6.8584	0.1000	0.0000	4.89	4.49	0.40
11	$(CH_3)(CH_3)N$—	2	SCH_3	7.3993	0.1000	0.0000	4.67	4.62	0.05
12	$(CH_3)(CH_3)N$—	2	OCH_3	6.4683	0.1000	0.0000	4.64	4.39	0.25
13	$(CH_3)(CH_3)N$—	2	CH_3	6.3560	0.1000	0.0000	4.44	4.37	0.07
14	$(CH_3)(CH_3)N$—	2	OH	6.0769	0.1000	0.0000	4.05	4.30	0.25
15	$(CH_3)(CH_3)N$—	2	H	5.9430	0.1000	0.0000	3.84	4.26	0.12

续表

编号	A_m	n	R	$^1\chi$	$^5\chi_{PC}$	$^6\chi_{CYT}$	$\lg\left(\frac{1}{c}\right)$① 实验	$\lg\frac{1}{c}$ 计算	$\lvert\Delta\lg\frac{1}{c}\rvert$
16	$(CH_3)_2N$—	3	Br	7.3502	0.1581	0.0000	4.12	4.23	0.11
17	$(CH_3)_2N$—	3	Cl	6.9601	0.1581	0.000	3.80	4.14	0.34
18		2	Cl	8.1955	0.0000	0.07905	4.88	4.42	0.46
19		2	H	7.6791	0.0000	0.07905	4.23	4.29	0.06
20		2	Cl	7.7728	0.000	0.0456	4.07	4.75	0.68
21				8.1706	0.0000	0.0000	5.20	5.45	0.25

① 用一种光电活动记录仪测量小鼠自发活动减低50%作为活性指标；c为受试物的浓度，单位为$mol\cdot L^{-1}$。

(4) 分子连接性指数与化合物其他的理化性质的关系研究　化合物分子的理化性质不仅取决于分子和原子的数目（加合性），而且还取决于原子在分子中的排列（构成性）。而分子连接性指数可以把分子的加合性和构成性的结构信息都译制出来，Kier、Hall、Murray等深入研究了分子连接性指数，认为其和分子若干的物理性质有显著相关性。例如，原子离解热和生成热、摩尔折射率、分子极化度、气体方程经验常数、气化热、沸点、液体的密度、有机液体的水溶性、分配系数等10余种物理性质观察值和用分子连接性指数与之建立的方程式而计算出来的数值几乎完全一致，读者可参阅文献[24]。

分子连接性法是1975年才提出来的，虽在不断发展中克服了不少问题。例如，初步解决了杂原子的点价等问题，但还有很多不足之处。例如指数的种类太多，物理意义不明确；指数值对几何异构体无法区别，所以顺反式的χ值是等同的；对化合物的构象也无法区别。另外，对磷、硼等元素在共价键中的δ值尚未规定出来。所以这些还有待于进一步发展、改进和完善。

二、基于模式识别的建模方法

将模式识别技术用于定量构效关系的研究是近年来取得的新进展。有关化学模式识别的一些方法在第七章已经介绍，在此只对它们在定量构效关系中的应用研究情况给出简要讨论。

化学学科的一个中心任务就是要研究化合物的结构与反应活性的关系，尤其是一些有关生化反应中的结构与活性的关系近年来受到化学家的广泛关注。正如本书前节讨论的那样，这些关系影响因素较多，如极性性质、立体效应等，用量子化学进行详细的理论计算则太复杂，而且计算量太大以致难于实施。这样，化学模式识别就成了现在少数几个有用的研究定量构效关系手段之一，值得引起化学家的重视[44]。

化合物的结构可由一个"主干"(backbone)和一些主干上特殊位置的取代基来描述[45]，这样一些特殊位置上的取代基的化学性质就可用在上节讨论的一些参数来进行描述，一些特殊的性质还可通过量子力学的计算得到，这样，一组具有相同化学结构主干的化合物就转变成为一组数据矢量来表示，不同的反应特性就应该具有不同的数据结构。一些已知反应特性的矢量数据可构成化学模式识别方法中的训练集（参见第七章）、通过训练，就可对一些具有同样化学结构主干的未知化合物的反应特性来进行预测了。

在这里讨论一个例子，有这样一系列具有如下化学结构主干的化合物：

$$X(Y)—C_6H_4\quad CH(R)—CH(R^1)—NH—R^2$$

在这里，5 种不同取代基 X、Y、R、R^1 及 R^2 可从 H 变化到 CH (CH_3) $CH_2C_6H_4$- OH，采用前述方法将这些化合物由一组数据矢量来表示，其中只有一个变量是由量测给出。这组数据的训练集由 32 个化合物所构成，每个化合物由 13 个变量表出。训练集由两类不同类型的化合物组成，一类的反应活性高，为兴奋剂，共 15 个；另一类的反应活性低，为阻滞剂，共 17 个。Dunn 等采用 SIMCA 方法对这些数据进行了分析，得到了一些有用的结果，并对未知物也作了预测，说明化学模式识别是一种有用的进行定量构效关系研究的方法[46]。类似的例子可参阅文献[47，48]。

钢材断裂性能的模式识别分析。瑞典钢铁公司试制了 15 个新钢种，在新钢种的钢材加工过程中，有 9 种钢材开裂，另 6 种不开裂，为了查明钢中微量元素对钢材开裂的影响，他们分析了这 15 种钢材中的 17 种微量元素，并用模式识别的 SIMCA 算法寻找规律、结果发现："好钢"的成分代表点集中在一个较小的区域，可包括在一个高维空间的包络面内；"坏钢"的数据点则很分散。这是因为：引起开裂的原因不止一种、故"坏钢"区事实上是多个区域的叠加，"坏钢"的样本点集没有包络面，Wold 称这类情况为"不对称 SIMCA 分析"[49]。

模式识别用于药物设计。据 1978 年估计，全世界用于找寻新药的费用每年达 20 亿美元左右，每发明一种重要的新药耗资为 4000 万美元。为了更快更省地开发新药物，迫切需要总结化合物分子结构和疗效的关系，以提高探索的命中率。这种构效关系研究可有演绎法和归纳法两种。演绎法是从量子生物学角度查明药物活性的机理从而确定何种结构最有效。但目前的知识水平距这一目标尚十分遥远。归纳法则是利用模式识别等方法从大量实验结果中总结规律。这一方法虽然是纯经验性质或半经验性质，但切实可行。由于新药研制中合成和药理工作量很大，费用也相当高，即使是模式识别方法的预测率的准确度不高，也能产生一定效益。用模式识别搞药物设计的最大困难，是如何选取有效的结构描述参数，一种常用的方法是假定生物活性与结构参数有线性关系，这当然是一种粗略的近似或简化。

Kowalski 和 Bender 用模式识别方法研究抗癌药的构效关系[50]。他们取 200 个化合物（其中 87 个有抗癌活性）。选择了 20 个结构参数，其中下列参数最为有效：

① 硫原子数/总原子数；

② C—S 键数/碳原子数；

③ S—H 键数；

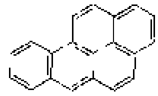
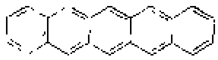
④ C—C 键数/碳原子数；

⑤ 碳原子数/总原子数。

用线性判别函数法和 K 最邻近法，分类率可达 90%左右。但也有人对这种分类判别方法提出不同意见。

Chu 等用线性判别函数法和 K 最邻近法用 51 个变量来识别 188 个药物的抗癌活性[52]。

预报 24 个未知化合物的抗癌活性，成功率达（83～92）%。

致癌物生物活性的模式识别研究。多环芳烃（PAH）致癌是有关环保和癌症病因的重要课题。人们通过大量实验注意到：多环芳烃的分子图形和致癌活性有很大关系。例如 3，4-苯并芘  是强致癌物质，而同样有 5 个苯环的  则不致癌。自从 Pullman 提出的 *K-L* 区理论以解释多环芳烃分子的结构与致癌活性关系以来，这一课题一直是理论生物化学注意中心之一。*K-L* 区理论和后来提出的湾区理论都注意到分子形状与致癌活性的关系。在遥感、指纹分类、汉字识别等领域广泛使用句法模式识别技术，是用计算机总结形状信息的有力工具，舒永昌和陈念贻就用这一技术研讨了多环芳烃致癌活性的规律[52]。

嗅觉活性化合物的模式识别研究。Schiffman 用模式识别方法研究香料化合物的构效关系。由 25 个结构化学参数（包括 Raman 光谱信息）组成的模式空间，用非线性映照的方法对 30 个化合物分类[53]。

Jurs 用线性判别函数法详细研究了麝香化合物的构效关系。训练集包括 60 个麝香化合物和 240 个非麝香化合物。用计算机产生 68 个结构描述单元（包括分子碎片、亚结构和几何描述单元），从中筛选出 13 个最有效的参数。计算结果表明：麝香化合物的香气质量不是由一个分子参数，而是由多个分子参数决定的。用 9 个未列人训练集的化合物的分类来检查模型的预报能力，分类效果很好[54]。

McGill 等用模式识别方法筛选影响化合物香味的主要结构因素，从红外光谱、紫外光谱、核磁波谱、分子量、熔点、沸点、密度、旋光性、水溶性、醇性和化学键参数中选择特征量，结果从 43 个变量中找到两个主要影响因素——分子的电子施主特性和偶极矩[55]。

微量元素与病因学的模式识别研究。近年来人们注意到，微量元素的比例失调是许多病（尤其是地方病）的病因或重要因素。微量元素硒的防癌作用近年来受到广泛注意，同时也发现其他几种元素对硒有拮抗作用。为了查明多种微量元素对癌症发病率的影响，陈念贻等[56]取 25 个国家和两个地区的居民（通过食物）对硒、锌、镉、铜、铬、砷的平均摄入量为模式识别的特征变量构成模式空间，将这些国家和地区的（根据年龄修正的）乳癌死亡率计入其中，作非线性映照，结果如图 9-6 所示，原始数据列于表 9-16。从图 9-6 可以看出，乳癌高发病国家（乳癌死亡率大于每 10 万人中 17 人）和乳癌低发病国家分布在不同区域，其间有明显的分界线。这一研究结果提出了一个发人深省的问题，能否调整微量元素的摄人量，使高发癌症国家转化为低发癌国家？

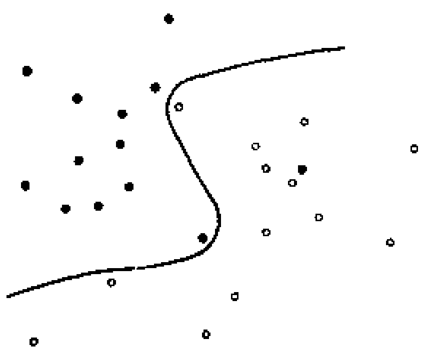


图 9-6 微量元素摄入量与癌死亡率的关系

●高摄入量、高死亡率国家和地区；○低摄入量、低死亡率国家和地区

模式识别与催化研究。催化问题是化工生产最重要的问题之一。90%以上的化工生产都要用催化剂。催化剂的研制往往是一个化工流程成败的关键。催化现象很复杂，广泛应用的多相催化尤其如此，从分子水平看，催化剂表面的原子排列和电子结构是催化作用的基础，由于催化剂成分很复杂，多为复相，表面结构和成分既未能完全查明，更无法用“原子级加工”来控制表面结构，活性中心往往由局部表面结构决定，而局部表面结构的量测至今尚无

妥善办法。另一方面，许多化学反应的机理也很复杂，不同的机理可导致不同的产物。催化剂表面结构与反应机理的关系是很难彻底查清的事。宏观的化学反应还牵涉到传热、传质和液体流动问题，这又和催化剂的显微结构有关。催化剂的制备方法、制备条件和显微结构有着复杂的联系……，所有这一切都决定着催化剂研究的复杂性。近年来，由于计算量子化学的长足发展，为我们提供了有关催化的大量信息，为从更深入的理论上解决催化问题投下了一线曙光，但真正用量子化学和表面分析技术彻底弄清和掌握催化问题还很遥远。在这种条件下，模式识别法倒可作出一些有益贡献。

表 9-16　元素摄入量与乳癌死亡率的关系

国家或地区	摄入量/($mg \cdot a^{-1}$)							死亡率
	Se	Cu	Zn	Cd	Cr	Mn	As	(1/100 000)
澳大利亚	75.9	1125	6948	123.9	21.1	722	152.4	19
奥地利	71.6	824	4272	77.0	22.0	858	102.4	17
比利时	70.6	741	4425	80.5	21.1	803	158.2	21
加拿大	61.8	874	5313	97.2	21.9	711	139.1	23.5
丹　麦	71.1	725	4387	76.9	18.6	677	166.0	24
前西德	64.3	784	4473	74.4	20.6	751	136.3	17.5
爱尔兰	75.1	836	3712	99.7	16.4	889	114.4	21.5
以色列	77.3	822	4449	77.1	23.2	923	136.9	21
荷　兰	57.8	693	3741	70.8	20.3	701	108.5	26
挪　威	82.2	608	4126	87.5	17.0	672	269.1	17.5
瑞　典	65.8	703	3783	74.8	18.3	646	167.1	18.5
瑞　士	65.6	850	4169	83.6	23.2	819	109.6	21.5
英　国	61.7	729	4502	85.4	17.7	717	132.2	25
美　国	61.0	849	5108	87.3	25.4	652	132.8	21.5
保加利亚	107.6	861	3924	79.5	16.7	1129	102.1	9.0
捷　克	85.1	838	4712	82.6	15.3	1029	120.2	15.5
芬　兰	67.1	690	3623	88.2	13.7	716	132.1	13
法　国	76.4	881	5339	91.5	21.2	965	173.8	16
希　腊	91.9	911	4452	92.3	17.7	1154	185.4	8.5
香港（中国）	91.2	639	2959	54.6	19.6	463	273.4	10
匈牙利	86.5	768	3958	72.4	14.9	1075	91.3	14.5
意大利	82.5	855	4304	83.8	16.4	1161	134.4	16.5
日　本	85.8	643	2420	43.4	17.4	674	233.4	3.5
波　兰	93.8	634	5231	80.3	15.9	876	138.3	11
葡萄牙	87.2	714	4347	74.6	15.8	1008	268.5	12.5
台湾（中国）	84.1	592	1674	33.3	12.8	522	184.3	4.0
前南斯拉夫	98.6	722	3288	72.9	11.7	1169	82.1	8.0

二维模式识别在催化研究中的应用。用两个变量作二维参数图有时也能总结一些催化规律。下面以乙炔与氯化氢气相加成反应为例加以说明。乙炔与氯化氢气相加成反应是基本有机合成的重要反应之一，用氯化物所含金属的电荷-半径比和电负性为坐标作图，20余种氯化物的催化活性在图上呈规律性分布（图 9-7）[57,58]。

从图 9-7 可以看出，Hg^{2+}、Pd^{2+}、Cu^{+}、Ag^{+} 氯化物的催化活性较高，其他阳离子活性较低，它们之间有明显分界线，界面方程为

$$z/r_k - 3x + 3.45 = 0$$

式中，z/r_k 表示电荷/半径，x 表示电负性。

此式可与刘祈涛提出的金属离子软硬酸度判据相对照

$$f = z/r_k - 3x + 2.2$$

由此可知，f 大于或小于 -1.25 是研究的催化活性高低的分类判据。这说明正是离子的软硬酸碱特性和这一反应有对应关系。

金属氧化物上的一氧化碳氧化反应是另一个例子。用金属氧化物的生成热 ΔH_0 和金属氧化物阳离子的半径 r 为坐标作图，一氧化碳氧化活性分布聚类较好（图 9-8）。活性低的 C 区有：BeO、GeO_2、Al_2O_3、SiO_2；活性高的 A 区有：MnO_2、CoO、MnO、CdO、Ag_2O、NiO、CuO、SnO_2、Cu_2O 和 ZnO，活性最高的氧化物，集中在 r 约为 65pm，$-\Delta H_0$ 在 209～376.2kJ・mol^{-1}范围。这暗示有几何因子和能量因子起作用。催化作用很复杂，用两个因素概括常常是不够的，因此需要多维空间模式识别。

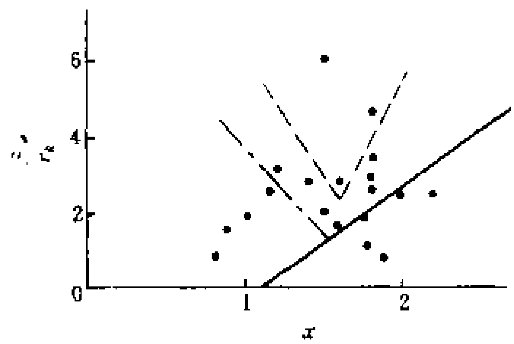


图 9-7　乙炔与氯化氢的加成反应中催化剂的活性

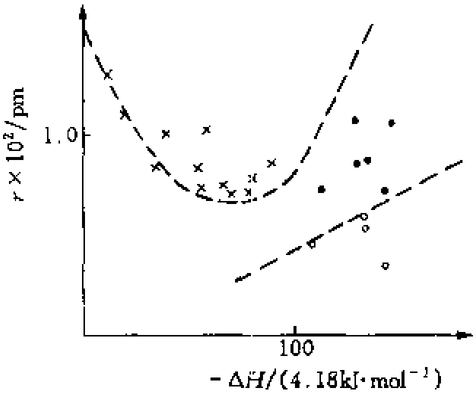


图 9-8　一氧化碳氧化反应催化剂活性

三、基于人工神经网络的建模方法

如在第六章和第七章所述，人工神经网络技术既可用于多元校正（回归）又可用于模式识别，特别是它的非线性映照的特性，更使得它在构效关系（QSAR）研究中独具魅力．故近年来在化学构效关系的建模中得到了十分广泛的应用，是目前化学计量学研究中的一个热点领域[59～62]。

人工神经网络在构效关系（QSAR）研究中有很好的应用前景。一些研究过的例子如 1-甲基-4-苯基 1,2,3,6-四氢吡啶（MPTP）的同类物对人肝脏二氢蝶啶还原酶（DHPR）的抑制作用的 QSAR[63]、硝基取代多环芳烃（NPAH）的致突变活性与分子结构参数之间的关系[64]、对位取代苯酚衍生物的生物活性与其结构及物理化学性质参数之间的关系[65]、氟络合物中 Eu(Ⅱ) 离子的 4f-4f 及 5d-4f 二类光谱跃迁与其结构参量之间的关系[66]，等等。这些课题均用人工神经网络方法取得了很好的结果。用人工神经网络处理传感器阵列数据，可进行化学物质[67]及其气味的辨识[68]。许多国家对投放市场的种子是否杂有外来异种有严格规定。Chtioui 等[69]比较了不同的人工神经网络方法用于种子的分类鉴别。Cinar[70]探讨了时间序列模型及人工神经网络用于动态化学过程模拟。基于径基函数的 ANN 模型被认为更适子描述动态化学过程。

总之，人工神经网络在构效关系研究中方兴未艾，可望解决一些在经典回归方法和模式识别方法难于解决的一些问题，是一个值得继续研究的领域。

参　考　文　献

1　Shorter J. Correlation Analysis of Organic Reactivity with Particular Reference to Multiple Regression. Chichester: Research

Studies Press, 1982

2 Hammett L P. Physical Organic Chemistry. New york: McCraw-Hill, 1940

3 Hansch C, Maloney P P, Fujita T et al. Nature (London), 1962; 194: 178

4 陈荣悌等. 化学学报. 1979; 37: 167; 1983: 41: 817; 1984; 42: 1044; 1985; 43: 884; 1986; 44: 424; 1988; 46: 757; 1988; 46: 808; 1990, 48: 349

5 蒋明谦. 有机化合物的同系线性规律. 北京: 科学出版社, 1980

6 徐光宪, 黎乐民. 中国科学, 1980; (2): 136

7 刘有成, 王云瑆. 化学学报, 1985; 43: 155; 1985: 43: 232

8 Exner O. Correlation Analysis in Chemistry, recent advances, Chapman N B et al. Eds., Chapter 10, Plenum, New York, 1978

9 Chapman N B, Shorter J. Advances in Linear Free Energy Relationships. New York: Plenum Press, 1972

10 Taft R W. Seperation of polar, steric, and resonance effects in reactivity. Melvin S. Newman (Ed). Steric effects in organic chemistry, New York: Wiley, 1956

11 Shorter J. The Separation of polar, steric, and resonance effects by the use of linear free energy in "Advances in linear free energy relationship" (Ed) Chapman N B. and J. Shorter. New York: Plenum Press, 1972

12 Hancock C K, Meyers E A, Yager B J. J Ame Che Soc. 1961; 83: 4211

13 Charton M. Journal of the American Chemical Society. 1969; 91: 3121

14 Kutter E, Hansch C. Journal of Medicinal Chemistry, 1969; 12: 647

15 Verloop A, Hoogenstraaten W, Tipker J. Development and applications of new steric substituent parameters in drug design in "Drug design, Vol Ⅶ" (Ed. by E. J. Ariens) New York: Academic Press, 1976

16 王尔华, 相秉仁. 国外医学药学分册, 1980; 3: 143

17 Pavelich W A, Taft R W. J Ame Chem Soc, 1957; 79: 4935

18 Pavelich W A. Thesis, Pennsylvania State University, 1955

19 Biechler S S, Taft R W. J Ame Chem Soc, 1957; 79: 5121

20 Bowden K, Chapman G, shorter J. J Chem Soc, 1963; 86; 5239; 1964; 87; 3370

21 Wiener H J. J Chem Soc, 1947; 69: 17～20

22 Barysz M, Plavsic D, Trinajtic N. MATCH, 1986; 19: 89～116

23 Balaban A T. J Chem Inf Comput Sci, 1992; 32: 23～28

24 Kier L B et al. Molecular connectivity in chemistry and drug research. 46. New York: academic Press, 1976

25 Randic M. J Chem Inf Comput Sci, 1984; 24: 301-310

26 Hansen P J. Jurs P C. J Chem Edu, 1988; 65: 574-580

27 Randic M. Brissey G M, Spencer R B et al. Comput Chem, 1980; 4: 27～43

28 Szymanski K, Muller W R, Knop J V et al. J Chem Inf Comput Sci, 1985; 25: 413～415

29 Randic M. J Chem Inf Comput Sci, 1986; 26: 134～136

30 H. P. Schultz. J Chem Inf Comput Sci, 1989; 29: 227～228

31 Muller W R, Szymanski K, Knop J V. et al. J Comput Chem, 1987; 8: 170～173

32 Schultz H P. Schultz E B, Schultz T P. J Chem Inf Comput Sci, 1990; 30: 27～29

33 Schultz H P, Schultz T P. J Chem Inf Comput Sci, 1991; 31: 144～147

34 Schultz H P, Schultz T P. J Chem Inf Comput Sci, 1992; 32: 364～368

35 Trinajstic N. Chemical Graph Theory; CRC: Boca Raton, FL, 1983; Vol. 1

36 Klavzar S. Gutman I. J Chem Inf Comput Sci, 1996; 36: 1001～1003

37 Plavsic D, Nikolic S, Trinajstic N, et al. Croat Chem Acta, 1996; 66: 345～353

38 Gutman I. Ruscic B, Trinajstic N et al. J Chem Phys, 1975; 62: 3399

39 Gorden M, Scantlebury G R. Trans Faraday Soc, 1964; 60: 605

40 Bonchev D, Balaban A T, Randic M. Int J Quantum Chem, 1981; 19: 61

41 Balaban A T. Pure Appl Chem. 1985; 55: 199

42 Murray W J. J Pharm Sci, 1977; 66: 1352

43 相秉仁等. 药学学报,1981; 16: 19
44 Kirschner G, Kowalski B R. The application of pattern recognition on drug design, in Drug Design. Vol. Ⅷ. New York: Academic Press, 1979
45 Cammarata A, Menon G K. J Med Chem, 1976; 19: 739
46 Dunn W J, Wold S. J Med Chem, 1978; 21: 1001
47 Norden B, Edlund U, Wold S. Acta Chem Scand Ser B, 1978; 21: 602
48 Dunn W J, Wold S. Bioorg Chem, 1980; 9: 505
49 Wold S. Anal Chim Acta, 1978; 103: 429
50 Kowalski B R et al. J Ame Chem Soc, 1974; 96: 916
51 Chu K C et al. J Medical Chemistry, 1975; 18: 539
52 舒永昌, 陈念贻. 分子科学学报,1981; 1: 39
53 Schiffman S S. Science, 1974; 185: 112
54 Jurs P C et al. J Agricultural and Food Chemistry, 1977; 25: 1158
55 McGill P et al. Anal Chem, 1977; 49: 596
56 陈念贻等. 计算化学及其应用,上海: 上海科技出版社,1987
57 白乃彬. 科学通报, 1978; 9: 544
58 白乃彬, 陈念贻. 自然杂志, 1980; 4: 272
59 Cao Z, Xu D, Jiang J H et al. Anal Chim Acta, 1996; 335: 117
60 Cao Z, Lin H G, Wang B F et al. Fresenius J Anal Chem, 1996; 355: 194
61 Jiang J H, Wang J H, Chu X et al. J Chemometrics, 1996; 10: 281
62 Jiang J H, Wang J H, Liang Y Z et al. J Chemometrics, 1996; 10: 241
63 Song X H, Yu R Q. Chemometrics Intell Lab Syst, 1993; 19: 101
64 Song X H, Xiao M, Yu R Q. Computers Chem, 1994; 18: 391
65 宋新华,陈茁,俞汝勤. 中国科学,1993; 23: 245
66 Song X H, Chen Z, Yu R Q. Chemom Intell Lab Sys, 1992; 16: 213
67 冯伟,胡上序. 高等学校化学学报,1996; 17: 1708
68 Song X H, Chen Z, Yu R Q. Computers Chem, 1993; 17: 303
69 Chtioui Y, Bertrand D, Devaux M F, et al. J Chemometrics, 1997; 11: 111
70 Cinar A. "Nonlinear time series models for multivariable dynamic processes", Paper of In CINC'94; URL: http://www.emsl.pnl.gov:2080/docs/incinc/ homepage. html

第十章 人工智能与化学专家系统

人工智能与化学专家系统是一个综合性很强的科技领域，可以说是本书中前述方法的一个逻辑性的集合。本书将其列为一章，其主要初衷是想对其在化学计量学研究中的地位给予必要的强调。另一方面，还因为人工智能与化学专家系统一般都需大量引用化学不同领域知识，需要建立数据库，又有其特殊性的一面，故将其分列成一章。本章将主要介绍一些常用的人工智能与化学专家系统的基本要素，在此基础上还将介绍一些如何构建化学专家系统的基本思路和框架。而对于如何具体构造数据库等细节知识将不在本书进行详细讨论，所以，本章将只有少量的计算机程序。因考虑到目前在化学研究领域中已有不少有名的专家系统，在本章将选择一些有代表性给出必要的说明，以供读者参考。

第一节 概 述

人工智能是一个综合性的科技领域。Hippe[1]建议将人工智能定义为求知能力的集合，这种能力是指获取知识并用于解决新的（未知的）理论和/或实际问题。而人工智能则是指借助技术或理论的手段，应用数学方法和/或数理逻辑研究智能的各个方面的问题[1]。简言之，人工智能就是研究如何用人工的方法与技术，即利用计算机等工具，模拟、延伸和扩张人的智能，实现某些机器思维或脑力劳动自动化[2]。人工智能是计算机和科学技术发展的前沿领域，化学计量学很自然地就成为它的应用领域之一。化学量测及其数据解析中，如何将各类分析仪器量测所得的数据转化为有用化学信息，传统上是依靠化学家、分析化学家运用其智能、专门知识、经验技巧及通过各类计算来完成的，能否设计计算机的专家系统，来模拟化学家和分析化学家的脑力劳动，这就是化学计量学所涉及的人工智能的任务。

本书前述的方法一般都是数值运算，而人工智能所涉及的计算机程序，则更多地涉及符号与逻辑处理过程。在化学领域中，如何从光谱图形（包括红外光谱、质谱、核磁共振谱等）出发，推断化合物的结构，这些问题远不是依靠单一数值运算可解决的问题。人工智能与一般数值运算的这一区别，表现在人工智能的计算机程序用的是启发式知识（推理）与启发式搜索方法，它借助于经验规则，或称“启发式”规则，以缩小搜索的范围。实际上，人类专家在处理问题时，正是这样做的。

人工智能的另一特征是大量引用领域知识。化学计量学中的人工智能所涉及的领域知识当然是化学的专门知识，这就涉及在建立数据库的基础上，再建立知识库、知识表达及推理方式。正如 Nilsson 所言[3]：“一个人工智能产生式系统的要素是：一个综合数据库，一组产生式规则和一个控制系统。”当前在人工智能的研究中，往往主张将数值方法与基于知识推理的逻辑方法和启发式搜索方法进行柔性集成，可望增强专家系统的解析功能。但在本章将主要介绍常用的搜索方法及有关知识表达与推理规则，并在此基础上介绍几个化学专家系统。

第二节 启发式分类与搜索方法

启发式知识的应用及启发式搜索是专家系统不同于其他数值计算（包括模式识别）的特征之一。启发式知识是一些不易精确地用数学语言描述的知识，化学专家系统用了化学家的

经验，例如谱图解析、结构识别、合成路线设计等均带有假设的色彩。启发式知识可定义为：有关目前问题状况与合适的解之间的经验性知识。而启发式分类是将不同分类层次上相互联系的概念，用不确定联想机制结合起来，启发式分类模型描述了专家系统知识和推理形式的特征，表达了经验性知识的组织和使用。

用人工智能解决一个问题，常可认作是在各种不同的可能性之间进行选择，或称“搜索”(search)，这个搜索空间可以用树的形式表述。图 10-1 就是这种搜索树[1]。

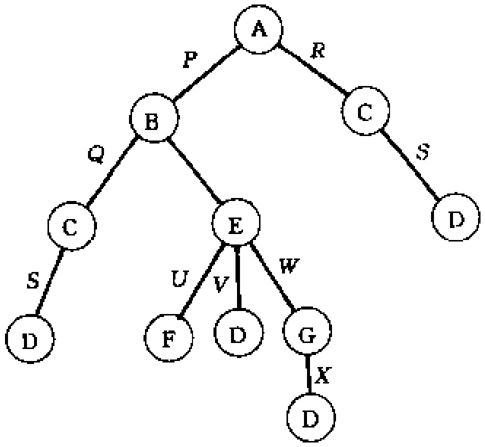


图 10-1　搜索树

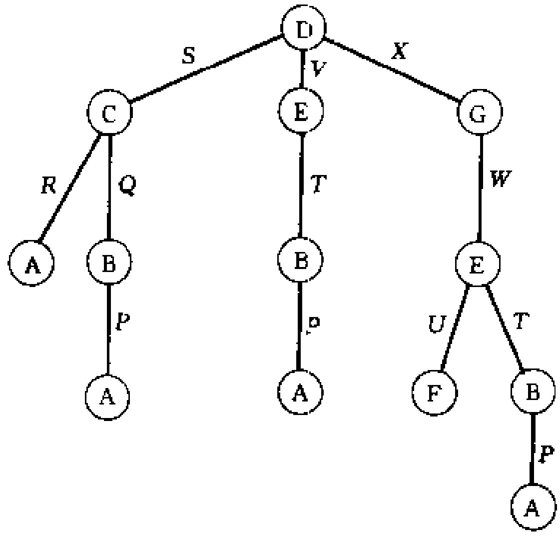


图 10-2　逆向搜索树

在化学中典型的问题是从化合物 A 出发，如何制取化合物 D？这可能有若干途径，但不可能由 A 直接制备 D，必须经过其他中间步骤，我们的目标是要找到最简捷的途径。图 10-1 可称为状态图，问题的解的路径是从初始状态（根结点）经过树的分支到达目标状态（终端结点）。对于大的实际问题，要绘出这样的显示图是十分困难的，因此，搜索树常是“隐式”的，在向目标搜索的进程中，计算机程序将自动产生各种可能的分支和结点，这里可分为正向推理与逆向推理两种情况，即在由起始状态 A 向目标状态 D 过渡时，图 10-1 是正向推理，但也可以逆向由目标状态朝起始状态搜索（图 10-2）。正向推理亦称是数据推动的，逆向推理又称目标引导的，在用人工智能辅助有机合成路线设计时，就采用逆向推理。对于很简单的问题，直接的方法是进行“盲目搜索”，这当然是很费时的办法，即让计算机依次产生并检验各结点可能出现的分支，随着搜索过程的进行，搜索树不断长大。以图 10-1 为例，每个结点可赋予一个水平，根结点位于 0 水平，接着是水平 1，余类推。水平的号次代表了搜索的深度。这里介绍两种盲目搜索方法。

一、广度优先搜索

这种搜索方法是从根结点出发，依次在每一水平生成并检验搜索树的各结点，在一个水平的各结点未检验完，即沿广度进行的“横向扫描”未完成前，不开始生成与校验更深水平的结点（即暂不“扩展”）。这里，认为同一水平各结点对问题的求解是等价的，只是按各结点生成的先后次序，先生成的先检验，沿“广度”遍历所有结点，然后才按原次序将先生成的结点先往后扩展，故称为广度优先搜索法。这种搜索法总是能找到以最简单步骤达到目标的路线，但不一定是最经济的解决问题的办法，这一点在下面讨论启发式搜索时将再论及。

二、深度优先搜索

此法进行搜索时，是由当前检验的结点作为父结点生成下一级子结点，这样由父结点生成子结点的过程继续下去，看每次检验是否出现目标结点，若未出现，再扩展最晚生成的子结点，如此下去，沿着最晚生成的子结点分支，逐级“纵向”深入发展，故称为深度优先搜索法。在到达顶点仍未找到目标结点时，往后“回溯”，在前一个分支处继续沿深度搜索。深度优先搜索不能保证一定找到最短路径的目标结点，因为如果目标结点不在最晚生成的子结点分支中，且该分支为无穷分支，则搜索过程将无限制地往下进行，即搜索失败。深度优先搜索法有可能在编制程序时只存储当前搜索的路径，对计算机内存要求较低。为改进深度优先搜索法，有人引入“搜索深度限制”，当沿“最晚”分支纵向搜索达到一定深度时，如仍未出现目标结点，则返回对“次晚”分支进行搜索，这种办法称有界深度优先搜索法。

上述讨论的盲目搜索法是正向推理，也可使用逆向推理，例如广度优先逆向推理法，对只要求少量搜索的问题，往往可用这种办法。

三、启发式搜索方法

总的说来，前面讨论的盲目搜索没有用到所涉及问题的领域知识来指导搜索。对于复杂的实际问题，如化学反应的设计，这种方法常常无法奏效，即由于排列组合太多，出现“组合爆炸”，即如果平均每个结点可有 n 个分支，则搜索空间将随深度 d 的增加按 n^d 关系增加。解决的办法是尽可能借助领域知识的帮助，这是人工智能提高求问题解的效率的基本诀窍。实际上，人们在日常生活中就是这样做的。虽这样得到的解不一定就是获得最优解的最佳途径，但总是在一步步接近目标。

前面叙述的广度优先搜索等方法，未考虑搜索过程中的“代价”问题，只要找到目标就行。如用加权树表示各支路的代价，这样可采用启发式的代价驱动搜索法来求解代价最小的路径，而运用启发性知识——所求解问题的有关知识和经验，如有机合成化学的知识，合成实验的化学专家的经验，可得到解的出现规律及解的某些性质，估计将要支付的代价等，为代价最小的路径求解带来很多启发性提示。运用启发式搜索的一种方法是局部择优搜索法，即搜索过程中根据领域知识，在一个局部点各可能路径中选择最有希望逼近目标结点的方向，例如爬山，选最陡上升方向爬，即梯度函数最大方向搜索，这在只有单峰极值的情况下能奏效。否则，可用全局择优搜索法，在同一级所有子结点中进行比较择优，同时，要对当前结点已付出的代价 g 及根据启发信息估计达到目标结点还需付出的代价 h 进行权衡，如 g 的权较大，搜索过程应倾向于广度优先搜索，强调横向扫描；如 h 的权较大，搜索过程应倾向于深度优先搜索，强调纵向深入。用这样的指导思想，能以较低代价、较高效率搜索到最优解。

在结束有关启发式分类与搜索的讨论之前，还要着重指出启发式是不确定的，基于典型性假设的、有时不易理解的一种关系。不确定性是因为启发式认识跳过了大量因果关系的中间层次，而这些中间层次在特定状况下可能不成立也可能造成错误判断，但正是跳过了中间层次，问题的求解才跨出了一大步。

第三节　知识表达技术

人工智能研究者早就发现，强的智能与其说是由于推理功能强，毋宁说是由于知识丰富。因此，人工智能将知识视为高效智能系统的关键要素，知识的表达与管理就成了人工智能研

究的中心课题之一。知识可定义为具有智能的人或机器能用以作出合理决策的信息。知识表达是指知识的表示与描述，亦即知识的形式化或模型化。

知识有叙述型、过程型与控制型等几种类型。叙述型知识叙述关于系统的状态、环境和条件，问题的构想、定义、事实等；过程型知识表述有关系统状态的变化、问题求解过程的操作等；控制型知识提供如何选择相应的操作、运算和行动的信息。可用于叙述型知识的表达方法有逻辑表达法、语义网络表达法等，而能用于表达过程型知识同时又能表达其他类型知识的表达方法，最典型的是产生式规则表达法。下面将举例简要介绍这三种知识表达方法。

一、逻辑表达法

下面是一个逻辑表达的例子：

① $\forall$ x[metal(x)→conduct(x)]；

② metal (iron)；

③ conduct (iron)。

此例中$\forall$ x [metal (x)→conduct (x)] 表示所有的金属都导电，metal (iron)；表示铁是金属，conduct (iron) 表示铁导电，全句意义为：所有的金属都导电，因为铁是金属，所以铁导电。在此，metal (x) 表示 x 是金属，是谓词逻辑中的一个原子谓词公式，x 是客体变元，它可以有一个定义域，称为客体域，而$\forall$ x (数学上表示任取 x) 称为“全称量词”，表示客体域中的所有个体。注意，一个谓词通常还可以有若干个客体变元，如谓词 p(x_1, x_2, …, x_n)，此时谓词 p 称为 n 元谓词。更复杂的情况还有，谓词 p (x) 中的变元 x 本身又可以为一谓词 (有点类似函数中的复合函数)，此时称 x 为一阶谓词，而 p (x) 则称为二级谓词。所以，知识的逻辑表达通常是指一阶谓词逻辑 (FOPL) 描述人工智能问题。谓词逻辑表达较严谨精确，具有通用性，接近自然语言，但使用效率低，灵活性较差，常使推理过程冗长。

二、语义网络表达法

语义网络表达是通过概念及其语义关系用网络图表达知识，用以描述样本、事件、概念等的关系。网络图实际上是图的一种，称有向图，有向图中从一个作为起点的结点到作为终点的结点的线称为弧。语义网络起初是为描述人类记忆的心理学模型提出的，在自然语言研究中，表示复杂句型的语义很方便。语义网络中结点表示对象、概念或事件，弧可用不同方式定义，这取决于被表示的知识，用于表达化学结构的普通弧有 isa 和 has-part，例如酮作为一种有机化合物概念的表述 (图 10-3)。注意，酮的特征基团羰基仅需在酮这一层上存储一次，而不是在各个特定的酮这样较低层次上重复存储。通过有关弧含义的知识，可以搜索网络，推出“丁酮含有羰基”这样的事实。这种表述法在推理过程涉及较复杂的推理时有用，它较自然，体现了联想思维过程，表达语义关系知识效率较高，但不及逻辑方法严谨，不便于表达判断性知识及动态知识。

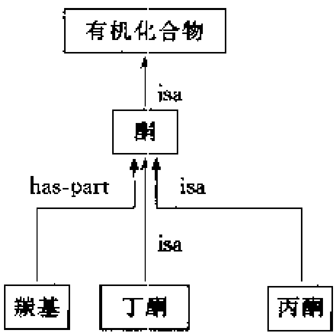


图 10-3　酮作为有机化合物概念的语义网络

三、产生式规则表达法

这是人工智能用于解决化学课题常用的知识表达方法，这种表达方法包含：

① 存储有关问题的状态、性质的叙述性知识的综合数据库，或称事实库；② 存储有关状态转移等规则的过程性知识的规则集，或称规则库；③ 用于选择控制策略，将规则与事实进行匹配、控制利用知识推理求解问题的控制器。

最简单的产生式规则有如下格式：

IF（如果）	THEN（则）
前提	结论
条件	行动

例如，在借助化学反应推测结构时，可将脱水反应表示为如下规则：

IF	存在模式 C—C—O
THEN	转换为模式 C═C

这是脱去一个 H_2O 的结构（H 原子均已略去）。产生式系统求解问题的过程，是使事实库转移到满足解的终止条件的状态。一条产生式规则的前提部分可以是能和事实库进行匹配的任何模式，若一条规则的前提部分被匹配，则该规则即属可用；使用一条规则的结果是得到一个结论或产生一个行动（如将 C—C—O 转化为 C═C），这将使事实库的状态发生转移。控制器按相应策略，控制规则与事实的匹配过程，有效地求解所涉及的问题。

产生式系统具有通用性，接近人的自然推理方式，易为用户理解。产生式规则可自由增删、修改，便于用户自行加入领域知识，但简单的产生式系统各规则独立，求解复杂问题时会出现“组合爆炸”，因而效率不高，表达能力亦不够强，需进行扩充，使其前提不限于简单的事实而是事实、情况和条件的综合模式。

除上述三种知识表达技术外，还有许多别的知识表述方法，如状态空间表达法，将有关初始状态、目标状态，及由前者达到后者所需的“操作”用符号形式表出，或用有向图（称状态空间图）表出；特征表表达法，即以表格形式表达样本特征，以用于检索；框架表达法，以各种大、中、小框架，互相内外嵌套组合表达状态变化与操作过程；与/或图表达法，这是一种超图，通常以树的形式，亦称与/或（AND/OR）树，这种表达知识的方法基于人们求解问题时的两种思维方法。AND 树，用于分解，将复杂的大问题分解成一组简单的小问题，将总问题分解为子问题，若所有的子问题解决了，总问题亦告解决，子问题解决不了的又可继续分解（图 10-4）；OR 树，用于变换，将较难的问题变换为较易的等价的问题，后者如能解决，原有难题亦告解决（图 10-5）。实际求解问题可兼用二法，即 AND/OR 树表达法。

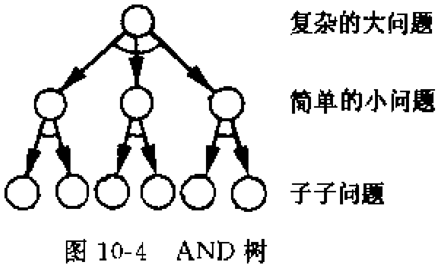


图 10-4　AND 树

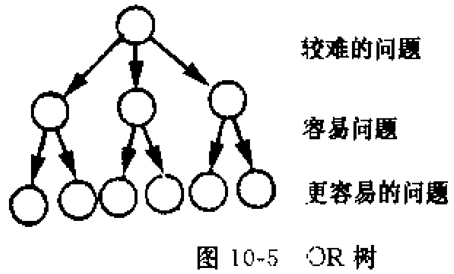


图 10-5　OR 树

知识表达仍是一个人工智能领域尚在继续研究完善的领域，如知识表达方式的标准化，不精确的知识的表达，表达效率的评估，知识的量化等方面都有待完善。

第四节　化学专家系统

一、化学专家系统的基本结构与特点

专家系统是人工智能的一个重要分支。从计算机人工智能的角度分析问题，任何用到大的化学知识库的复杂的人工智能系统都可认作化学专家系统，也就是说，用于解决实际化学问题的人工智能系统均是化学专家系统，可能“化学知识系统”是一个更确切的名词。

专家系统在化学中的应用极其广泛，在谱图解析、分离科学、分析方法的选择、仪器控制等方面都产生了很多有名的化学专家系统（参见表10-1）。

表10-1　国内外较知名的化学专家系统

系统名称	作　者	国　别	内　容	文　献
DENDRAL	J. Lederberg	美国	谱图解析，MS，^{13}C-NMR	5，6
CHEMICS	佐佐木慎一等	日本	谱图解析，MS，^{13}C NMR，^{1}H NMR，IR	7
CASE	M. E. Munk 等	美国	谱图解析，^{13}C NMR，IR	8
PAIRS	H. Woodruff 等	美国	谱图解析，IR	9，10
STREC	L. A. Gribov 等	前苏联	谱图解析，MS，^{13}C-NMR，^{1}H NMR，UV 等	11
CARBON	J. Zupan 等	南斯拉夫	谱图解析，^{13}C-NMR	12
	卢佩章等	中国	色谱专家系统	16，17
ESESOC	许禄等	中国	谱图解析，^{13}C NMR，MS，IR	13～15
PLATO	Curry 等	美国	谱图解析，^{13}C NMR，MS，IR	18
LHASA	W. Kaufman	荷兰	有机合成系统	19

开发始于20世纪60年代的DENDRAL可说是专家系统这一人工智能的主流分支诞生的一个标志，它由美国斯坦福大学创建，是一个有机分析化学系统，可从量测到的有机化合物的质谱数据出发，一直到产生出化合物的结构图。现以这一化学专家系统为例，说明化学专家系统的基本结构。一般说来，一个化学专家系统应包括如下组成部分：

（1）化学知识库（或化学知识源）　包括相关化学领域的事实、与问题有关的启发推理等。DENDRAL系统的知识库中包括从实验测得的质谱数据导出的有关有机化合物分子结构的限制的规则（哪些结构可能存在，哪些结构可能不存在）；产生能满足这些限制的可能结构的方法；从结构预测其质谱的规则等。最常用的表达化学知识的方法是前述产生式规则。

（2）推理步骤或控制结构　用于利用化学知识解决相关问题。DENDRAL系统用向前搜索系统。

（3）工作内存区或称“全局数据库”　存储所处理的问题的当前信息，包括输入的数据、已得到的结果等。DENDRAL系统工作时的全局数据库，是质谱数据、产生的结构限制、可能的结构等。

从化学专家系统的结构，可以看出它明显不同于一般前述的化学计量学数值计算方法之处，是将有关的一般化学知识、规则与当前课题的信息（输入数据），以及用一般知识解决当前问题的方法（规则解析器）明确分开，这样，化学专家系统易于接受新的化学知识并适应新的情况，以解决新的问题。化学专家系统的基本结构示意列于图10-6。

要构造一个化学专家系统，还必须满足下述前提：① 至少目前有化学专家能很好地解决涉及的问题；② 化学专家解决涉及的问题应是来源于特殊知识、判断能力与经验；③ 化学专家应能解释他所用的特殊知识、经验诀窍和解决问题的方法；④ 涉及的任务应有明确的应用

对象与范围．由于构造一个专家系统相当费时，这一考虑是十分必要的。化学家和计算机专家合作构造的解析质谱及其他波谱的DENDRAL等化学专家系统，即是在具备相应前提条件下构造的。

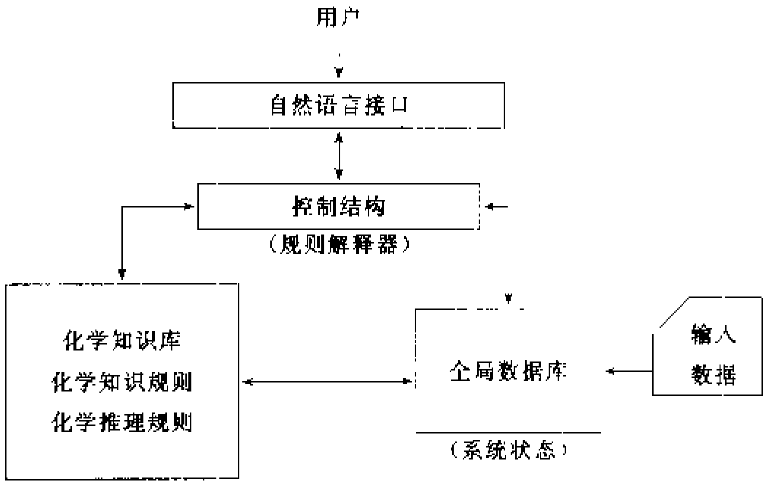


图10-6　化学专家系统的基本结构

以下将简要介绍几个化学专家系统。

二、DENDRAL质谱、核磁共振谱图解析专家系统

DENDRAL质谱、核磁共振谱图解析专家系统由诺贝尔奖获得者Lederberg主持开发，可以说是化学专家系统，甚至是专家系统这一人工智能的主流分支诞生的一个标志。它主要包括3个子系统：

① 启发式DENDRAL系统；

② 结构生成CONGEN系统；

③ “Meta-DENDRAL”系统。

诺贝尔奖获得者Lederberg曾致力于研究用拓扑学表征化学结构[6]。一个给定的化学成分，有多少可能的结构？Lederberg曾用LISP语言编制递归算法研究了从原子集合产生所有的非环形分子结构。启发式DENDRAL系统就包括了这些非环形和环形分子结构的穷举程序，其数据调整器从质谱图中检出对结构生成有重要意义的质谱峰；预推理器根据质谱峰进行初步分析，判断化合物属于哪一类；结构生成器根据预推理器输出的信息，生成对应于质谱图的所有可能的分子结构，将其列入可能分子结构候选表；预测器给出候选结构的预测质谱图，然后将预测质谱图与原谱图比较，检查二者符合情况，按评价函数决定其符合程度，以便按符合程度输出候选分子结构。

结构生成CONGEN（限制发生器）系统，于20世纪70年代中期设计，以取代原DENDRAL系统的非环形和环形的分于结构的穷举程序，所以，CONGEN是一个功能更强的限制性的分子结构生成器。通过人-机交互，使用CONGEN的化学家可以随时方便地输入启发信息，对分子结构的生成过程加人约束条件。CONGEN还增加了对立体异构的考虑。新的CONGEN程序系按深度优先搜索编制，并允许化学家使用时提前停止计算。

“Meta-DENDRAL”系统是一个能够自动建立化学知识库的学习系统，这对本身理论尚不完善的质谱解析来说有特别意义。在质谱结构解析专家系统构造中，如何使质谱解析专家的知识和经验明确化、系统化，表述为产生式规则，本身就是一个难题。“Meta-DENDRAL”系

统通过向样品质谱学习，自动进行归纳处理。

例如，一个简单的质谱裂解规则 R1 为

$$\mathrm{R1：N—C—C—C \longrightarrow N—C^{*}C—C}$$

① 寻找分子中所有与这条规则左边匹配的分子子图部分；

② 对于每个匹配，将分子在规则右边加星号标记的键断开；

③ 保留星号左边的部分（例如保留 N—C），并记录保留部分的质量。

在一个大分子里，规则 R1 可多次使用，例如，$CH_3—CH_2—CH_2—NH—CH_2—CH_2—CH_2—CH_3$ 的谱线，包含数据点质量在 72 和 86，使用这个规则导出两个分裂碎片：

$$CH_3—CH_2—CH_2—NH—CH_2\cdot$$

$$\cdot CH_2—NH—CH_2—CH_2—CH_2\quad CH_3$$

“Meta-DENDRAL”系统通过学习，从已知化合物的质谱图归纳出有机化合物的裂解规则，学习过程示于图 10-7。

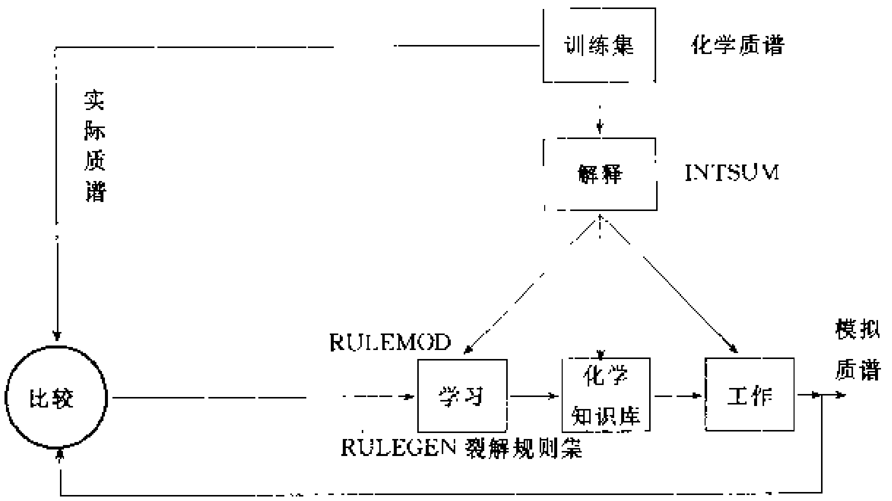


图 10-7 Meta-DENDRAL 的学习过程

首先，由解释程序 INTSUM 从训练集总结出初步的裂解规则，每个分子通过寻求能够解释每个质谱峰产生原因的一个或多个裂解过程，为该分子说明其质谱中的每个峰，利用下述限定条件可限制应取的裂解过程数目：

第一，只考虑其裂解碎片与质谱峰的质量数相对应的断裂。

第二，质谱领域的语义模型限制，即只考虑半阶理论允许的裂解。

在质谱领域知识的语义模型中，有所谓零阶理论，这是指分子内部每个键的子集都能断裂，但这个理论不足以限制搜索范围，因此，把某些通用的指导方针强加在它上面，称半阶理论。这一理论断言在质谱分析过程中，一些将要断开面且原子将迁移，关于裂解有如下限制：① 双键和三键不断开；② 芳烃键不断开；③ 在数据中只应出现比两个碳原子还大的裂解碎片；④ 相同碳原子的两个键不能同时断开；⑤ 在任一次裂解中，断开的键不能多于三个；⑥ 在一个过程中不会出现多于两次的完全裂解；⑦ 在多步过程中最多只有两个环形裂解碎片。关于原子迁移有如下规则：⑧ 在一次裂解之后，至多有两个氢原子能够迁移；⑨ 在任一次裂解之后，至多脱除一个 H_2O；⑩ 在任一次裂解之后，损失至多不过一个 CO 单位。

语义模型是化学与质谱学基本知识及专家经验的总结。利用语义模型，化学质谱专家可以灵活地修订增删约束条件，指导规则生成过程。在 INTSUM 总结出初步裂解规则之后，“工作”环节根据裂解规则形成模拟质谱，“比较”环节将模拟质谱与实际质谱进行比较，驱

动“学习”环节。学习环节中有 RULEGEN（规则产生）与 RULEMOD（规则修正）两个程序。RULEGEN 寻找较 INTSUM 产生的初步裂解规则更一般的规则集，它不仅要符合已有的样本，还可解释新的质谱数据，即一般的规则应能正确解释质谱中很多数据点（正的证据），也可预测一些在任意图谱中都不出现的峰（负的证据）。正的证据是成功的标志，由这些规则引进的负的证据则由 RULEMOD 通过规则合并、消除冗余等方法进行修正。假设某一次已成功地使用了一次规则，一旦这个规则中还有一个不合适的例子，RULEMOD 能适当地修改这个规则。这里系统的输入是样本质谱与语义模型，而输出的是关于分子结构分裂过程的产生式规则集。最后，将得到的裂解规则存储于知识库。

可以看出，上述学习过程体现出专家系统程序向训练集样本学习，而同时也吸收了化学与质谱专家的知识与经验。

“Meta-DENDRAL”的应用，不但确证了一些过去已发现的质谱测定法的规则，而且还发现了一些过去未报道的甾烷分子的新规则。“Meta DENDRAL”系统亦用于^{13}C-NMR 的解析。

三、PLATO 数据解析专家系统

PLATO 数据解析专家系统（GC/IR/MS）由 Curry 等开发[8]，其结构框图如图 10-8 所示，其系统主要由四部分所组成：

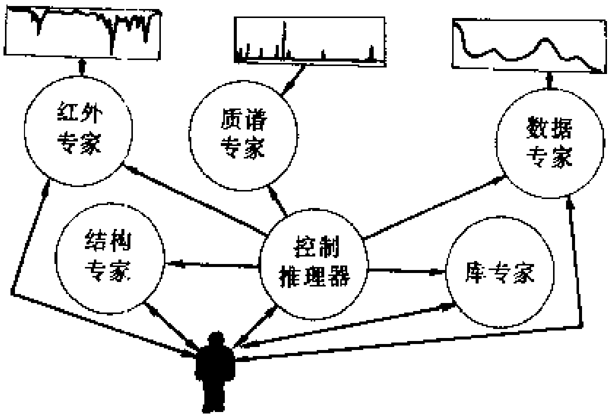


图 10-8 PLATO 系统框图

图中圈表示程序模块

① 控制-推理器模块；

② 数据库管理模块；

③ 分子结构编辑模块；

④ 数据专家模块。

这个系统主要用于 GC/IR/MS 数据的自动解析，其中“数据专家”主要模拟谱图学家收集和解析数据；“控制器”则在系统中起化学家所起的作用；“库管理”是对谱图库中的谱图、结构及子波谱进行管理和检索；而“结构编辑”则是完成结构的分解、装配和显示的任务。

该系统是一个分布式专家系统，其程序模块可以并行运行，故可加快解析速度；另外，该系统可对各种不同类型的数据进行处理，且可根据不同的数据类型，选用不同的数据分析算法。该系统允许用户干预或控制某一问题解析的全过程，如光谱的归属、官能团约束以及于结构的装配，且系统与用户间的交互作用是完全遵循人类专家演绎推理的过程和原则来进行的。

（一）数据专家

该系统的数据专家为一独立模块，主要由三部分组成：① 红外“数据专家”（IR expert）；② 质谱“数据专家”（MS expert）；③ 数据库专家（data expert）。此三部分子模块可进行相互交换与连接，还可相互调用。

（二）控制器

控制器是 PLATO 的心脏，它的功能就像一个化学家，如设定分析的目的，组织“数据专家”，并把“数据专家”所得结果进行综合，以求得问题的最终答案。

（三）知识库

在PLATO系统中，知识库以层次方式组织了1000个化学子结构，知识库中的每一项为一“化学类”，它们包含了最重要的有机官能团，该知识库分布在所有PLATO的模块中，但维护和更新则统一由控制器进行。

知识库包含了所有重要的与红外光谱相关的子结构，同时也包含了质谱和核磁共振谱系统中的主要子结构。谱图的手段不同，对于子结构的表达之侧重亦各不相同，如红外所侧重的是官能团，质谱侧重的为元素的组成，而^{13}C-核磁共振谱（NMR）则侧重碳原子的局部化学环境。对系统的查询，所得结论以及对结论的解释均以这些子结构为基础。

四、高效液相色谱专家系统

此高效液相色谱专家系统由中国科学院大连化学物理研究所开发[16,17]，是一个与色谱仪器直接联用的集成系统，由以下几个子系统所组成：

① 全自动化高效液相色谱仪；

② 高效液相色谱样品预处理及最佳柱系统推荐模块；

③ 高效液相色谱最佳分离条件优化模块；

④ 高效液相色谱谱图库及其验证模块；

⑤ 高效液相色谱定性定量数据处理模块。

这些子系统既各自自成体系，相互之间又可连接和相互交流，是一个有机的整体[20,21]。该系统从20世纪80年代初开始开发，历经10年方得以完成。

该系统的作者认为，一个完整的高效液相色谱专家系统应当包括样品预处理方法推荐、最佳柱系统推荐、操作条件优化以及色谱数据的定性定量，而且更为重要的是这些模块之间有机联系。作者指出，某些复杂样本需作适当预处理，使得其变得容易分离和检测，进而使得推荐的柱系统更佳，而只有当推荐的柱系统最佳，继而进行的操作条件优化才有真正的价值；反过来，也只有是在最佳条件比较不同的柱系统所得结果才有效，故预处理、柱系统选择和操作条件优化是不可完全分立的。如果还更深层次考虑，色谱的定性库不仅可用于谱图库中谱峰的定性，还可以用于条件优化，再进一步来比较不同方法之间的差异，筛选出最佳方案来改善柱系统的推荐。所以，上述专家系统的几个子系统必须有机地结合起来，在充分发挥它们各自功能的基础上，还必须进行全面合理安排。一方面需要一个总体控制模块来控制选择调用专家系统中的每一个子系统，另一方面还需考虑每一模块之间的相互连接，使每一模块之间能够相互调用，共享数据库和谱图库[22]。

我们将该专家系统的各模块称为子系统是因为前述的各模块都可相应的看成一个独立的专家系统，因为它们各自都具有专家系统所需的各要素，即它们都具有自己的化学知识库（或化学知识源），自己的推理步骤或控制结构以及数据库等。有关该系统的详细介绍，读者可参阅文献[23]。

五、ESESOC有机化合物结构解析专家系统

该系统由中国科学院长春应用化学研究所开发[13~15]，其结构框图如图10-9所示，其解析过程从逻辑上可分为三步：

① 由实验数据（如质谱、核磁共振波谱和红外光谱等）或者由化学信息（如分子式）出发，在知识库（如谱图-结构相关规则）中获得化合物结构碎片集；

② 在结构碎片集的基础上，利用知识库（如诸多约束条件），经结构发生器进行整体结构的对接，生成各种可能的分子异构体，得候选化合物；

③ 在波谱模拟、分子张力能计算，^{13}C 谱峰信息，模式识别及人机交互信息等的作用下，进行候选化合物的验证，直至将最可能的结构输出。

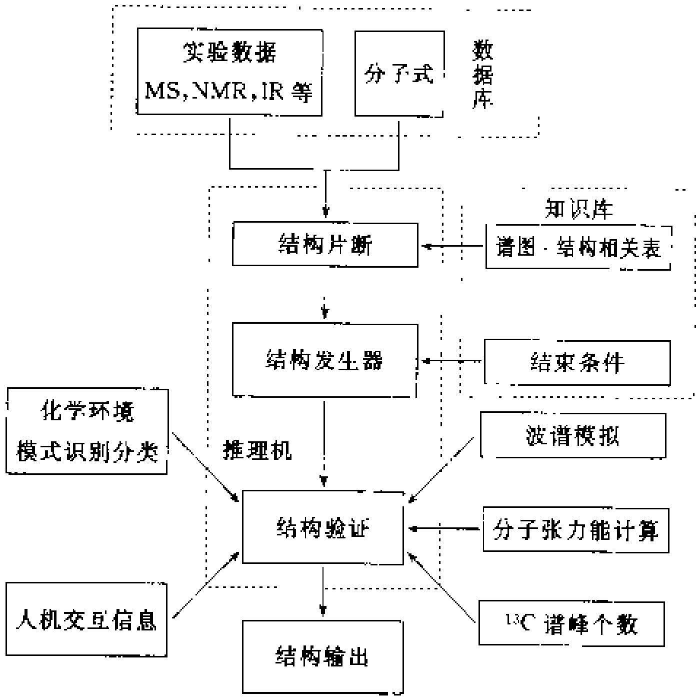


图 10-9　ESESOC 系统的基本结构

在上述步骤中，他们认为核心部分是结构发生器，因为必须要求结构发生器所产生的候选化合物是遍历所有可能结构，亦即穷举且无冗余的。在此步中需灵活运用图论、离散数学和拓扑学等手段，是该系统较难的一部分。另一方面，因随着化合物原子数的增大，还将遇到“组合爆炸”的问题，使穷举难于实现，确为此类系统的难点问题。

有关该系统的详细信息，可参阅文献[14]。

参 考 文 献

1　Hippe Z. Anal Chim Acta，1983；150：11

2　涂序彦. 人工智能及其应用. 北京：电子工业出版社，1982

3　Nilson N J. Principle of Artificial Intelligence，Tioga Publishing Co.，1980

4　Gevarter W B. Artificial Intelligence，Expert Systems. Computer Vision and Natural Language Processing. Noyes Publishing，N. J.，1984

5　Lindsay R K，Buchanan B G，Feigenhaum E A et al. Applications of Artificial Intelligence for Organic Chemistry　The Dendral Project. NewYork：McGraw-Hill，1980

6　Lederberg J et al. J Am Chem Soc，1969；91：2973

7　Sasaki S，Kuoo Y，J Chem Inf Comput Sci，1985；25：252

8　Shelly C A，Munk M E. Anal Chim Acta，1981；133：507

9　Tomellini S A，Stevenson J M，Woodruff H B. Anal Chem，1984；56：67

10　Tomellini S A，Hartwixk R A. Anal Chim Acta，1984；162：227

11　Gribov L A，Elyyasberg D. CRC Crit Anal Chem，1979；8：111

12　Zupan J，Novic M，Baanec S et al. Anal Chim Acta，1987；200：333

13　许禄. 分析化学，1990：18：180

14 许禄，郭传杰. 计算机化学方法及应用. 北京：化学工业出版社，1990

15 Zhang J G, Xu L. Anal Chim Acta, 1988; 210: 163

16 卢佩章等. 色谱，1990; 8: 133; 1990; 8: 201; 1990; 8: 215; 1990; 8: 270

17 Lu P, Huang H. J Chromatogr, 1990; 452: 285

18 Curry B. "A distributed expert system for interpretation of GC/IR/MS data", Meuzelaar (Ed.), HLC. in "Computer-Enhanced Analytical Spectroscopy" Vol 2. NewYork: Plenum Press, 1990

19 Baw A Feigenbaum A E. Ed, Handbook of artificial Intelligence. Vol 1. Kaufman W In., 1981; Vol 2. 1982

20 Lu P, Huang H. J Chromatogr, 1988; 452: 172

21 梁鑫淼，张玉奎，卢佩章. 第七次全国色谱学术报告会论文集，北京：556

22 梁鑫淼. 博士学位论文，1992

23 卢佩章，张玉奎，梁鑫淼. 高效液相色谱法及其专家系统. 沈阳：辽宁科学技术出版社，1994

第十一章　统计学和应用数学基础知识

因本书讨论的有关化学计量学的方法用到不少线性代数和统计学知识，而且有些知识在一般数学教科书上还很难找到，为使本书成为一个封闭系统，故将一些必要的线性代数和统计学知识收录于此，汇成一章供读者参考。

在介绍这些知识之前，先讨论一下化学量测数据的矢量和矩阵表示方法。在分析化学的测量中，一个化学样本一般都可由一个谱来表示，如常见的光谱，色谱，各类波谱和各种电化学谱。将这些谱离散化，就可得到一组数据，这组数据就相当于数学中的矢量。如果我们把很多样本收集在一块，就可得到一个矩阵。一般说来，在这样的数据矩阵中，对一个样本量测所得到的一个谱可置为一个行矢量，这样的行矢量可称为一个样本对象或一个样本；与之类似，在这样的数据矩阵中，一个变量（如一个分析通道，一个波长等）可置为一个列矢量，这样的列矢量可称为一个变量（variable）。本书所论及的大多数化学计量学方法，都是建立在这样的矢量和矩阵之上的，所以有关矢量和矩阵运算的线性代数基础知识对化学计量学就显得特别重要了。另一方面，由于在分析量测中总得与量测误差打交道，怎样来估价这些随机误差对量测数据的影响，这就得需要统计学知识，在本书讨论的是多变量化学计量学方法，特别需要多元统计知识。

第一节　统计学基础知识

一、随机事件的概率公式

（一）随机事件及其概率

在一定的条件下，现象 A 可能发生，也可能不发生，我们把发生了现象 A 的事件叫做随机事件 A，简称事件 A。如果在既定的条件下进行一组试验，总共试验 N 次，其中现象 A 发生了 N_A 次，则该组试验中事件 A 的频率为 (N_A/N)，重复进行很多组这样的试验，可发现随机事件的频率具有某种规律性，如果从统计意义上说，随机事件的频率存在着一个极限值，称为事件 A 的概率，记为 $P(A)$，

$$\lim_{N\to\infty} (N_A/N)=P(A)$$

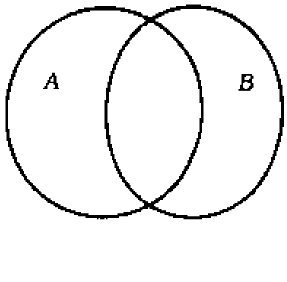


图 11-1　事件的和

若事件 A 和事件 B 是两个不同的随机事件，A 和 B 的和事件 $(A+B)$ 是指 A 与 B 中至少有一个发生的事件。如用两个圆分别表示事件 A 和事件 B 的集合，如图 11-1 所示，则两个圆的总和就代表 $A+B$ 的集合。显然，只有 A 发生，或者只有 B 发生，或者 A、B 同时发生的事件都是事件 $(A+B)$。

定义事件 A 与事件 B 的积事件 AB 为 A 和 B 同时发生的事件。图 11-2 中两个事件重叠的区域就是积事件的区域。

（二）和事件概率公式

和事件 $A+B$ 的概率就是图 11-1 的区域中事件集合的概率，其概率公式为

$$P(A+B)=P(A)+P(B)-P(AB)$$

如果 A 和 B 不可能在一次试验中同时发生，即

$$P(AB)=0$$

则称事件 A 与事件 B 是互斥事件（参见图 11-3）

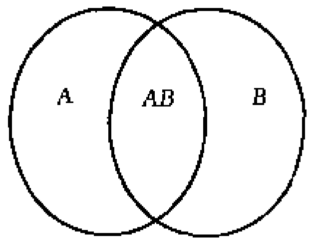


图 11-2　事件的积

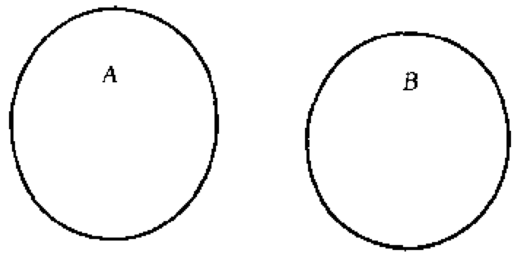


图 11-3　互斥事件

（三）积事件概率公式和条件概率

设 B 为任意事件，$P(B)>0$，则在 B 发生的条件下，事件 A 的条件概率定义为

$$P(A|B)=\frac{P(AB)}{P(B)} \tag{11-1}$$

即

$$P(AB)=P(B)P(A|B)$$

同理有

$$P(AB)=P(A)P(B|A) \tag{11-2}$$

上式（11-2）即称为积事件的概率公式。

对于事件 A 和事件 B，如果 A 事件的概率不受 B 是否发生的影响，即

$$P(A|B)=P(A) \tag{11-3}$$

则称事件 A 独立于事件 B。显然，对于相互独立的事件 A 和 B，积事件概率公式为

$$P(AB)=P(A)P(B) \tag{11-4}$$

（四）全概率公式

符号 $\{A_i\}$ 表示某一事件组 $\{A_1,A_2,\cdots\}$。如果任意一次试验的结果都至少发生 $\{A_i\}$ 中的一个事件，则称 $\{A_i\}$ 是一个事件的完备集。显然，对于事件的完备集 $\{A_i\}$，有

$$P(\sum A_i)=1 \tag{11-5}$$

式中，$\sum$ 表示对 $\{A_i\}$ 中的所有事件求和。

如果 $\{A_i\}$ 是一个互斥事件的完备集，式（11-5）可直接写成

$$P(\sum A_i)=\sum P(A_i)=1 \tag{11-6}$$

而对于任一随机事件 B 有下列全概率公式

$$P(B)=\sum P(BA_i)=\sum P(B|A_i)P(A_i) \tag{11-7}$$

这是因为 $\{A_i\}$ 是一个互斥事件的完备集，而 B 总是伴随 $\{A_i\}$ 中的现象同时发生，所以，事件 B 可以表示成式（11-7）所示的互斥积事件之和。

$$B=BA_1+BA_2+\cdots=\sum(BA_i) \tag{11-8}$$

（五）贝叶斯公式

事件 A 对于事件 B 的条件概率，可以由事件 A 和事件 B 的概率以及 B 对于 A 的条件概率算出，这就是关于条件概率的贝叶斯（Bayes）定理：

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)} \tag{11-9}$$

如果 $A=\{A_i\}$，而 $\{A_i\}$ 是一个互斥事件的完备集，利用全概率公式（11-8），公式（11-9）又可写成

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}=\frac{P(B|A)P(A)}{\sum[P(B|A_i)P(A_i)]}$$

二、随机变量及其分布

（一）随机变量

对于任一随机实验，它的样本空间是 $S=\{e\}$，如果对于每一个属于该样本空间的样本有一个实数 $X(e)$ 与之相对应，这样就得到一个定义上的实数单值函数 $X(e)$，则 $X(e)$ 称为随机变量。

有些随机变量，它全部可能取到的值是有限个或是可列无限多个，这种随机变量称为离散型随机变量，否则称为连续型随机变量。

设 X 是一个随机变量，x 是任意实数，函数 $F(x)=P\{X\leqslant x\}$ 称为分布函数。在这里 $P\{X\leqslant x\}$ 表示随机变量 X 取值小于 x 的概率。对于任意实数 $x_1, x_2(x_1\leqslant x_2)$，有

$$P\{x_1\leqslant X\leqslant x_2\}=P\{X\leqslant x_2\}-P\{X\leqslant x_1\}=F(x_2)-F(x_1) \tag{11-10}$$

因此，若已知 X 的分布函数就能知道 X 落在任一区间（x_1，x_2）上的概率。从这意义上来说，分布函数完整地描述了随机变量的统计规律性。

分布函数具有以下的基本性质：

① $F(x)$ 是一个不减函数。事实上，

$$F(x_2)-F(x_1)=P\{x_1\leqslant X\leqslant x_2\}\geqslant 0 \qquad (x_1\leqslant x_2)$$

② $0\leqslant F(x)\leqslant 1$　这是因为概率的最大取值为 1。

一般，如果对于随机变量 X 的分布函数 $F(x)$，存在非负的函数 $f(x)$，使对于任意实数 x 有

$$F(x)=\int f(t)\mathrm{d}t \tag{11-11}$$

则 X 称为连续型随机变量，其中函数 $f(x)$ 称为 X 的概率密度函数，简称概率密度。很明显，连续型随机变量的分布函数是连续函数。由定义知道概率密度具有以下性质：

① $f(x)\geqslant 0$

② $\int f(x)\mathrm{d}x=1$

③ $P\{x_1\leqslant X\leqslant x_2\}=F(x_2)-F(x_1)=\int f(x)\mathrm{d}x$

④ 若在 x 点处连续，则有　　$\mathrm{d}[F(x)]/\mathrm{d}x=f(x)$

（二）几种最重要的连续型随机变量的分布

1. 均匀分布

设连续型随机变量 X 在有限区间（a，b）内取值，且其概率密度为

$$f(x)=\begin{cases}1/(b-a) & a\leqslant x\leqslant b\\ 0 & \text{其他}\end{cases}$$

则称 X 在区间（a，b）上服从均匀分布。

在区间（a，b）上服从均匀分布的随机变量，具有下述意义的等可能性，即它落在区间

(a, b) 中任意长度的子区间内的可能性是相同的，或者说它落在子区间内的概率只依赖于于区间的长度而与子区间的位置无关。事实上，对于任意长度为 h 的子区间

$$(c, c+h), a\leqslant c\leqslant c+h\leqslant b,$$

有

$$P\{c\leqslant X\leqslant c+h\}=\int f(x)\mathrm{d}x=\int 1/(b-a)\mathrm{d}x=h/(b-a)$$

均匀分布在计算机数值模拟中具有相当重要的地位。

2. 正态分布

设连续型随机变量 X 的概率密度为

$$f(x)=(1/\sqrt{2\pi}\sigma)e^{-(x-\mu)^2/(2\sigma^2)} \qquad -\infty\leqslant x\leqslant\infty \tag{11-12}$$

式中，$\mu, \sigma\geqslant 0$ 为常数，则称 X 服从参数为 μ, σ 的正态分布或高斯分布，记为 $X\sim N(\mu, \sigma^2)$。而 μ 和 σ 分别称为正态分布的数学期望和均方差，是两个非常重要的参量。它的图形示于图 11-4。特别是当 $\mu=0, \sigma=1$ 时，称 X 服从标准正态分布。

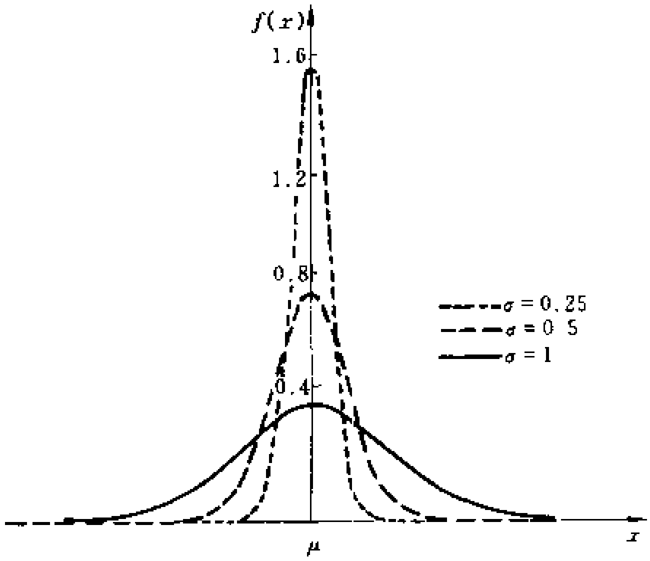


图 11-4　不同 σ 的正态分布图

$f(x)$ 的图形对称于 $x=\mu$，σ 的大小影响图形的形状，σ 大则图形矮胖，σ 小则图形瘦高。

一般正态分布可以通过适当变换化成标准正态分布。

若 $X\sim N(\mu, \sigma^2)$，$Z=(X-\mu)/\sigma$，则 Z 服从标准正态分布，即 $Z\sim N(0, 1^2)$。

3. χ^2 分布

设连续型随机变量 X 的概率密度为

$$f(x)=\begin{cases}\{1/[\Gamma(n/2)(2^{n/2})]\}x^{(n/2-1)}e^{-x/2} & x>0\\ 0 & x\leqslant 0\end{cases}$$

式中，$\Gamma(\cdot)$ 为伽玛函数，n 为正整数，则称 X 服从自由度为 n 的 χ^2 分布，记为 $X\sim\chi^2(n)$。对不同的自由度 n，$f(x)$ 的图形示于图 11-5。χ^2 分布是一个十分重要的分布，这是因为存在有如下一个定理：

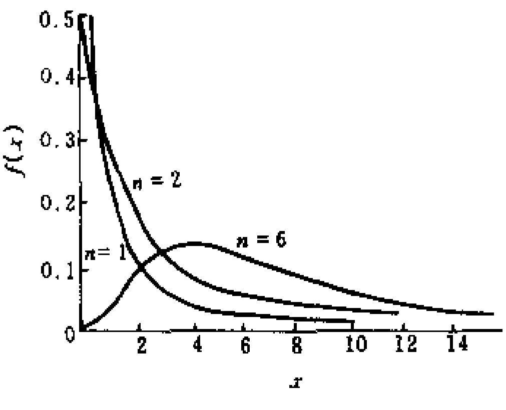


图 11-5　不同自由度 (n) 的 χ^2 分布图

若 $\xi_1, \xi_2, \ldots, \xi_n$ 是相互独立并且都服从正态分布 $N(0,1)$ 的随机变量，则随机变量

$$\chi^2=\xi_1^2+\xi_2^2+\cdots+\xi_n^2 \tag{11-13}$$

服从自由度为 n 的 χ^2 分布。

当 $n=1$ 时，$\chi^2(1)$ 分布又叫 Γ 分布，当 $n=2$ 时，$\chi^2(2)$ 分布就是指数分布。

$\chi^2(n)$ 分布的可加性定理　若 $\chi_1^2\sim\chi^2(n_1)$，$\chi_2^2\sim\chi^2(n_2)$，且相互独立，则

$$\chi_1^2+\chi_2^2\sim\chi^2(n_1+n_2) \tag{11-14}$$

χ^2 分布表及其用法　对给定的 $\alpha(0<\alpha<1)$，若有一点 $\chi_\alpha^2(n)$ 满足条件

$$P\{\chi^2>\chi^2_\alpha(n)\}=\int f(x)\mathrm{d}x=\alpha \qquad (11\text{-}15)$$

则称此点为 α 分位点。对于不同的 $\alpha,n,\chi^2(n)$的数值已制成数表，称为 $\chi^2(n)$分布表，可在任一本统计学书上方便查到。

4. t 分布

（1）定义　t 分布的概率密度函数定义为

$$f(t)=\{\Gamma[(n+1)/2]/[\Gamma(n/2)\sqrt{n\pi}]\}(1+t^2/n)^{-(n+1)/2} \qquad (-\infty<t<\infty) \quad (11\text{-}16)$$

式中，$\Gamma(\cdot)$为伽玛函数，n 为正整数，是 z 自由度参数。式（11-16）表示随机变量服从自由度为 n 的 t 分布，记为 $T\sim t(n)$。注意到由于 $f(t)=f(-t)$，所以它的图形关于纵轴对称，当 n 增大时，它将近似于正态分布，如图 11-6 所示。

定理：若随机变量 X 和 Y 相互独立，且 X 服从正态分布 N（0，1），Y 服从自由度为 n 的 χ^2 分布，则

$$T=\frac{X}{\sqrt{\frac{Y}{n}}} \qquad (11\text{-}17)$$

服从自由度为 n 的 t 分布。

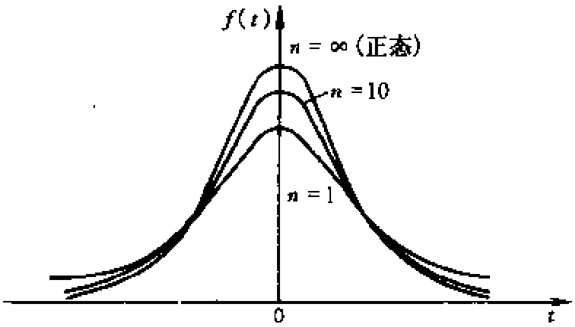


图 11-6　不同自由度的 t 分布图

（2）$t(n)$分布表及其用法　对给定的 $\alpha(0<\alpha<1)$，若有一点 $t_\alpha(n)$满足条件

$$P\{T>t_\alpha(n)\}=\int f(t)\mathrm{d}t=\alpha$$

则称此点为 $t(n)$分布轴上的 α 分位点。由于 $f(t)$ 图形的对称性，有

$$t_\alpha(n)=t_{1-\alpha}(n)$$

对于不同的 $\alpha,n,t_\alpha(n)$的数值已制成数表，称为 $t(n)$分布表，可在任一本统计学书上方便查到。表中所列 n 最大为 45，当 $n>45$ 时，就用 $N(0,1^2)$分布来近似替代。

5. F 分布

（1）定义　若随机变量 η 和 ξ 相互独立，且分别服从自由度为 m 和 n 的 χ^2 分布，则随机变量

$$F=(\eta/m)/(\xi/n)$$

的分布密度函数为

$$f_F(x)=\begin{cases}\{[(m/n)^{m/2}x^{m/2-1}]/\mathrm{B}(m/2,n/2)\}[1+(m/n)x]^{-(m+n)/2} & x>0 \quad (11\text{-}18a)\\ 0 & x\leqslant 0 \quad (11\text{-}18b)\end{cases}$$

式中，$\mathrm{B}(\cdot,\cdot)$为贝达（Beta）函数，其定义为：若 p，q 为正实数，函数

$$\mathrm{B}(p,q)=\int x^{p-1}(1-x)^{q-1}\mathrm{d}x \qquad (11\text{-}19)$$

称为贝达（Beta）函数。随机变量 F 称为服从自由度（m，n）的 F 分布，简记为 $F\sim F(m,n)$。注意，F 分布有两个自由度，第一个自由度 m 是组成 F 分布的分子随机变量的自由度，第二个自由度 n 是组成 F 分布的分母随机变量的自由度，密度函数 $f_F(x)$的图形示于图 11-7。

（2）F分布的性质

① 若$X \sim F(m,n)$，则$\frac{1}{X} \sim F(n,m)$，

② $F_{1-\alpha}(n,m)=\frac{1}{[F_{\alpha}(m,n)]}$

(3) F分布表及其用法　对给定的$\alpha(0<\alpha<1)$，若有一点$F_{\alpha}(n,m)$满足条件

$$P\{F>F_{\alpha}(n,m)\}=\int f(x)\mathrm{d}x=\alpha$$

则称此点为$F_{\alpha}(n,m)$分布的上α分位点。对于不同的$\alpha,n,m,F_{\alpha}(n,m)$的数值已制成数表，称为$F(n,m)$分布表，可在任一本统计学书上方便查到。注意到$F(n,m)$分布表中所给的$\alpha$都是很小的数，如0.10、0.05、0.01等等，当α较大时，如$\alpha=0.95$，表中查不到，这时根据前述F分布的性质②，就可查到。比如$F_{0.95}(15,12)=1/[F_{0.05}(12,15)]=1/2.48=0.403$。

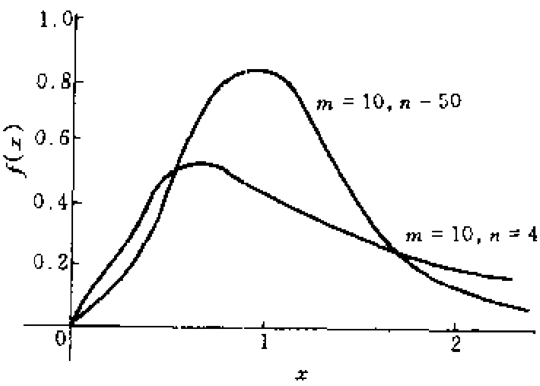


图 11-7　不同自由度的F分布图

三、随机变量的数值特征

(一) 数学期望

设离散型随机变量X的分布律为$P\{X=x_k\}=p_k,(k=1,2,\cdots)$，若级数$\sum x_k p_k$绝对收敛，则称级数$\sum x_k p_k$为$X$的数学期望，记为$E(X)$，即

$$E(X)=\sum x_k p_k$$

对于连续型随机变量，若它的概率密度为$f(x)$，注意到$f(x)\mathrm{d}x$的作用与离散型随机变量中的p_k相类似，于是自然有以下的定义：

设连续型随机变量X的概率密度为$f(x)$，若积分$\int f(x)\mathrm{d}x$绝对收敛，则称积分$\int f(x)\mathrm{d}x$为X的数学期望，记为$E(X)$，即

$$E(X)=\int f(x)\mathrm{d}x \tag{11-20}$$

数学期望简称期望或称为均值。数学期望具有以下性质：

① 设c是常数，则有　　$E(c)=c$。

② 设X是一个随机变量，c是常数，则有　　$E(cX)=cE(X)$

③ 设X,Y是任意两个随机变量，则有　　$E(X+Y)=E(X)+E(Y)$

④ 设X,Y是两个相互独立的随机变量，则有$E(XY)=E(X)E(Y)$

(二) 方差

设X是一个随机变量，若$E\{[X-E(X)]^2\}$存在，则称$E\{[X-E(X)]^2\}$为X的方差，记为$\mathrm{Var}(X)$或$D(X)$，即

$$\mathrm{Var}(X)=D(X)=E\{[X-E(X)]^2\} \tag{11-21}$$

在应用上还引入与随机变量具有相同量纲的量$\sqrt{\mathrm{Var}(X)}$，记为$\sigma(X)$，称为标准差或均方差。

对于离散型随机变量，按上述定义，则有

$$\mathrm{Var}(X)=D(X)=\sum[x_k-E(X)]^2 p_k \tag{11-22}$$

其中，$P\{X=x_k\}=p_k,(k=1,2,\cdots)$。

对于连续型随机变量，则有

$$\mathrm{Var}(X)=D(X)=\int[X-E(X)]^2 f(x)\mathrm{d}x \tag{11-23}$$

式中，$f(x)$为 X 的概率密度为 $f(x)$。关于随机变量的方差的计算有如下公式：

$$\mathrm{Var}(X)=D(X)=E\{[X-E(X)]^2\} \tag{11-24}$$

根据前述数学期望的性质，容易证得

$$\begin{aligned}\mathrm{Var}(X)=D(X)&=E\{[X-E(X)]^2\}\\&=E\{X^2-2XE(X)+[E(X)]^2\}\\&=E(X^2)-E[2XE(X)]+E\{[E(X)]^2\}\\&=E(X^2)-2E(X)E(X)+[E(X)]^2\\&=E(X^2)-[E(X)]^2\end{aligned}$$

同样利用数学期望的性质，容易证明方差的以下性质：

① 设 c 是常数，则有 $\mathrm{Var}(c)=0$。

② 设 X 是一个随机变量，c 是常数，则有 $\mathrm{Var}(cX)=c^2\mathrm{Var}(X)$

③ 设 X，Y 是两个相互独立的随机变量，则有 $\mathrm{Var}(X+Y)=\mathrm{Var}(X)+\mathrm{Var}(Y)$

第二节　应用数学基础知识

一、矢量及其运算

1. 列矢量与行矢量

当说到一个矢量 $\boldsymbol{a}$ 时，一般是指列矢量，行矢量 $\boldsymbol{a}^t$ 是指它的转置，数学表达式如下：

$$\boldsymbol{a}=\begin{bmatrix}a_1\\a_2\\\vdots\\a_n\end{bmatrix}\qquad \boldsymbol{a}^t=[a_1,a_2,\cdots,a_n]$$

两个矢量说成相等，当且仅当它们的每一个元素都对应相等。即如果

$$\boldsymbol{a}=\begin{bmatrix}a_1\\a_2\\\vdots\\a_n\end{bmatrix}\qquad \boldsymbol{b}=\begin{bmatrix}b_1\\b_2\\\vdots\\b_n\end{bmatrix}$$

只有当 $a_i=b_i(i=1,2,\cdots,n)$时，$\boldsymbol{a}$ 和 $\boldsymbol{b}$ 才相等。

2. 矢量加减法

两个或两个以上的矢量相加(或相减)，是指它们的对应元素相加(或相减)，即

$$\boldsymbol{a}+\boldsymbol{b}=\begin{bmatrix}a_1&+&b_1\\a_2&+&b_2\\\vdots&&\vdots\\a_n&+&b_n\end{bmatrix}$$

矢量的加减法具有以下性质：

$$\boldsymbol{a}+\boldsymbol{b}=\boldsymbol{b}+\boldsymbol{a}\quad\text{(交换律)}$$

$$(\boldsymbol{a}+\boldsymbol{b})+\boldsymbol{c}=\boldsymbol{a}+(\boldsymbol{b}+\boldsymbol{c})\quad\text{(结合律)}$$

$$\boldsymbol{a}+\boldsymbol{0}=\boldsymbol{a}$$

在这里

$$\mathbf{0}=\begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}$$

称为零矢量，它的每一个元素都等于 0。

对于任一矢量 $\boldsymbol{a}$，都存在一个与之相反的矢量 $-\boldsymbol{a}$，它们的和等于零矢量，即

$$\boldsymbol{a}+(-\boldsymbol{a})=\mathbf{0}$$

3. 矢量的数乘

如果 k 是一个实数（或称标量），则它与任一矢量 $\boldsymbol{a}$ 的乘积为该矢量的每一元素与它的乘积，即

$$k\boldsymbol{a}=\begin{bmatrix}ka_1\\ka_2\\\vdots\\ka_n\end{bmatrix}$$

矢量的数乘具有以下性质：

$$k_1(k_2\boldsymbol{a})=(k_1k_2)\boldsymbol{a} \qquad \text{（结合律）}$$

$$k_1(\boldsymbol{a}+\boldsymbol{b})=k_1\boldsymbol{a}+k_1\boldsymbol{b} \qquad \text{（分配律）}$$

$$(k_1+k_2)\boldsymbol{a}=k_1\boldsymbol{a}+k_2\boldsymbol{a} \qquad \text{（分配律）}$$

对于任一矢量 $\boldsymbol{a}$ 有

$$0\boldsymbol{a}=\mathbf{0};\ 1\boldsymbol{a}=\boldsymbol{a};\qquad -1\boldsymbol{a}=-\boldsymbol{a}$$

4. 矢量的内积（或称点积）和外积

两个矢量的内积产生一个数，即

$$\boldsymbol{a}^t\boldsymbol{b}=[a_1,\ a_2,\ \cdots a_n]\cdot\begin{bmatrix}b_1\\b_2\\\vdots\\b_n\end{bmatrix}=\sum a_ib_i$$

很明显，

$$\boldsymbol{a}^t(\boldsymbol{b}+\boldsymbol{c})=\boldsymbol{a}^t\boldsymbol{b}+\boldsymbol{a}^t\boldsymbol{c} \qquad \text{（分配律）}$$

$$(\boldsymbol{a}+\boldsymbol{b})^t\boldsymbol{c}=\boldsymbol{a}^t\boldsymbol{c}+\boldsymbol{b}^t\boldsymbol{c} \qquad \text{（分配律）}$$

如果两矢量的点积为零，则称它们相互正交，即 $\boldsymbol{a}^t\boldsymbol{b}=0$。

两个矢量的外积产生一个矩阵，即

$$\boldsymbol{a}\boldsymbol{b}^t=\begin{bmatrix}a_1\\a_2\\\vdots\\a_n\end{bmatrix}[b_1,b_2,\cdots,b_n]=\begin{bmatrix}a_1b_1 & a_1b_2 & \cdots & a_1b_n\\a_2b_1 & a_2b_2 & \cdots & a_2b_n\\\vdots & \vdots & & \vdots\\a_nb_1 & a_nb_2 & \cdots & a_nb_n\end{bmatrix} \qquad (11\text{-}25)$$

这样的矩阵可称为双线性矩阵，在多元分辨中有着特殊的地位。

二、矩阵及其运算

$n\times m$ 个元素排成一个矩阵：

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

称为 n 行 m 列矩阵，或称 $n\times m$ 阶矩阵。元素 a_{ij}可以是数，也可以是代数式。当 $n=1$ 时，称这矩阵为 m 维行矢量；当 $m=1$ 时，称这矩阵为 n 维列矢量；$n=m$ 时，称为方阵。

通常用大写黑体字母 $\boldsymbol{A}$，$\boldsymbol{B}$，…表示矩阵，用带有下标 i，j 的小写字母 a_{ij}，b_{ij}，…分别表示矩阵 $\boldsymbol{A}$，$\boldsymbol{B}$，…的第 i 行第 j 列的元素，因此，有时也用（a_{ij}）表示矩阵 $\boldsymbol{A}$，矩阵 $\boldsymbol{A}$ 有时也可以用列矢量来表示，即，$\boldsymbol{A}=[\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_m]$。

1. 矩阵的加减法

两个或两个以上的具有相同阶数的矩阵相加（或相减），是指它们的对应元素相加（或相减），即

$$\boldsymbol{A}-\boldsymbol{B}=(a_{ij})+(b_{ij})=(a_{ij}-b_{ij})$$

矩阵的加减法具有以下性质：

$$\boldsymbol{A}+\boldsymbol{B}=\boldsymbol{B}+\boldsymbol{A} \qquad (\text{交换律})$$

$$(\boldsymbol{A}+\boldsymbol{B})+\boldsymbol{C}=\boldsymbol{A}+(\boldsymbol{B}+\boldsymbol{C}) \qquad (\text{结合律})$$

2. 矩阵的数乘

如果 k 是一个实数（或称标量），则它与任一矩阵 $\boldsymbol{A}$ 的乘积仍为一矩阵，该矩阵的每一元素由 $\boldsymbol{A}$ 中相应的元素与 k 的乘积所构成，即，$k\boldsymbol{A}=k\ (a_{ij})\ =\ (ka_{ij})$。容易证明，矩阵的数乘具有以下性质：

$$k_1(k_2\boldsymbol{A})=(k_1k_2)\boldsymbol{A} \qquad (\text{结合律})$$

$$k_1(\boldsymbol{A}-\boldsymbol{B})=k_1\boldsymbol{A}+k_1\boldsymbol{B} \qquad (\text{分配律})$$

$$(k_1+k_2)\boldsymbol{A}=k_1\boldsymbol{A}+k_2\boldsymbol{A} \qquad (\text{分配律})$$

3. 矩阵乘矩阵

$(n\times q)$阶矩阵 $\boldsymbol{A}=(a_{ij})_{n\times q}$与$(q\times m)$阶矩阵 $\boldsymbol{B}=(b_{ij})_{q\times m}$的积是一个$(n\times m)$阶矩阵 $\boldsymbol{C}=(c_{ij})_{n\times m}$，它的元素 c_{ij}由等式

$$c_{ij}=\sum a_{ik}b_{kj}$$

决定，即为 $\boldsymbol{A}$ 的第 i 行与 $\boldsymbol{B}$ 的第 j 列的点积。注意到矩阵乘矩阵时一般不满足交换律，即

$$\boldsymbol{AB}\neq\boldsymbol{BA}$$

但是满足结合律：

$$\boldsymbol{ABC}=(\boldsymbol{AB})\boldsymbol{C}=\boldsymbol{A}(\boldsymbol{BC})$$

同样也满足分配律：

$$\boldsymbol{A}(\boldsymbol{B}+\boldsymbol{C})=\boldsymbol{AB}+\boldsymbol{AC}$$

$$(\boldsymbol{A}+\boldsymbol{B})(\boldsymbol{C}+\boldsymbol{D})=\boldsymbol{A}(\boldsymbol{C}+\boldsymbol{D})+\boldsymbol{B}(\boldsymbol{C}+\boldsymbol{D})$$

4. 零矩阵和单位矩阵

零矩阵用 $\boldsymbol{0}$ 表示，它的全部元素都为 0。一个 n 阶方阵，假如其对角线上的元素都是 1，非对角线上的元素都是 0，就称为单位矩阵。它可用 $\boldsymbol{I}$ 或 $\boldsymbol{I}_n$ 表示，很明显，$\boldsymbol{0}$ 和 $\boldsymbol{I}$ 有下列两个类似于数 0 和数 1 所具有的性质：

$$\boldsymbol{A}+\boldsymbol{0}=\boldsymbol{A} \tag{11-26a}$$

$$\boldsymbol{IA}=\boldsymbol{AI}=\boldsymbol{A} \tag{11-26b}$$

5. 矩阵的转置

把矩阵的 A 行和列互相调换位置后，所得的矩阵 A^t 称作矩阵的转置矩阵，也即若 A^t 的第 i 行第 j 列的元素用 $a_{ij}{}^t$ 表示，A 的第 i 行第 j 列的元素用 a_{ij}表示，则有关系式

$$a_{ij}{}^t = a_{ji} \tag{11-27}$$

由以上式（11-27）定义，可得到如下的关系式：

$$(\boldsymbol{AB})^t = \boldsymbol{B}^t\boldsymbol{A}^t \tag{11-28a}$$

$$(\boldsymbol{ABC})^t = \boldsymbol{C}^t\boldsymbol{B}^t\boldsymbol{A}^t \tag{11-28b}$$

如果矩阵 A 的转置矩阵就等于其本身，即 $A^t = A$，则称该矩阵为对称矩阵。

6. 逆矩阵

如果两个同阶方阵 A 和 B，满足 $AB = I$，那么 B 是 A 的逆阵并记为 A^{-1}，如果 A^{-1}存在，则称 A 是非奇异矩阵，或称为满秩矩阵，否则称是奇异矩阵。矩阵的逆运算具有以下性质：

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1} \tag{11-29}$$

$$(\boldsymbol{A}^t)^{-1} = (\boldsymbol{A}^{-1})^t \tag{11-30}$$

7. 矩阵的行列式

方阵的行列式是一个数，记为 $\det(A) = \alpha$，α 为任一实数。如果为一 n 阶方阵，有

$$\det(\boldsymbol{A}) = \sum (-1)^{i+j} a_{kj} \det(\boldsymbol{A}_{kj})$$

其中 A_{kj}是$(n-1)\times(n-1)$阶矩阵，它是划去的第 k 行和第 j 列所得到的 A 的子阵。有用的行列式性质包括

$$\det(\boldsymbol{AB}) = \det(\boldsymbol{A})\det(\boldsymbol{B})$$

$$\det(\boldsymbol{A}^t) = \det(\boldsymbol{A})$$

$$\det(k\boldsymbol{A}) = k^n \det(\boldsymbol{A}) \qquad \boldsymbol{A} \text{ 是}(n\times n)\text{阶矩阵}$$

$$\det(\boldsymbol{A}) \neq 0 \quad \Leftrightarrow \quad \boldsymbol{A} \text{ 是非奇异矩阵}$$

8. 正交矩阵

如果一个方阵 A，满足 $A^tA = I$，那么则称 A 为正交矩阵，显然有

$$\boldsymbol{A}^t = \boldsymbol{A}^{-1}$$

$$\det(\boldsymbol{A}) = \pm 1$$

这是因为

$$\det(\boldsymbol{A})\det(\boldsymbol{A}) = \det(\boldsymbol{A}^t)\det(\boldsymbol{A}) = \det(\boldsymbol{A}^t\boldsymbol{A}) = \det(\boldsymbol{I}) = 1$$

9. 方阵的迹

一个方阵 A 的迹，记为 $\mathrm{tr}(A)$，定义为它的主对角线上元素的和，即

$$\mathrm{tr}(\boldsymbol{A}) = \sum a_{ii} \tag{11-31}$$

特别是当 A 是个一行一列的矩阵，即为一个数 a 时，有

$$\mathrm{tr}(a) = a$$

这一性质在一些证明中经常用到，例如二次型 y^tAy 是一个数，所以

$$\mathrm{tr}(\boldsymbol{y}^t\boldsymbol{A}\boldsymbol{y}) = \boldsymbol{y}^t\boldsymbol{A}\boldsymbol{y} \tag{11-32}$$

方阵的迹具有一些性质：

$$\mathrm{tr}(\boldsymbol{A}+\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{A}) + \mathrm{tr}(\boldsymbol{B}) \tag{11-33a}$$

$$\mathrm{tr}(\alpha\boldsymbol{A}) = \alpha\,\mathrm{tr}(\boldsymbol{A}) \tag{11-33b}$$

$$\mathrm{tr}(\boldsymbol{AB}) = \mathrm{tr}(\boldsymbol{BA}) \tag{11-33c}$$

$$E[\mathrm{tr}(\boldsymbol{A})]=\mathrm{tr}[E(\boldsymbol{A})] \tag{11-33d}$$

在此，E [tr($\boldsymbol{A}$)] 表示对矩阵 $\boldsymbol{A}$ 的迹求数学期望（或称均值），式（11-33d）所示的性质表示求迹运算与求数学期望的运算可交换。数学期望的定义已在统计学基础知识中介绍。

10. 矩阵的秩

一个 $n\times m$ 阶的矩阵 $\boldsymbol{A}$，其秩是 $\boldsymbol{A}$ 中最大线性无关行数（或列数），记为 rank（A）。它具有下述性质

$$0\leqslant \mathrm{rank}(\boldsymbol{A})\leqslant \min(n,m) \tag{11-34a}$$

$$\mathrm{rank}(\boldsymbol{AB})\leqslant \min[\mathrm{rank}(\boldsymbol{A}),\mathrm{rank}(\boldsymbol{B})] \tag{11-34b}$$

$$\mathrm{rank}(\boldsymbol{A}+\boldsymbol{B})\leqslant \mathrm{rank}(\boldsymbol{A})-\mathrm{rank}(\boldsymbol{B}) \tag{11-34c}$$

$$\mathrm{rank}(\boldsymbol{A}'\boldsymbol{A})=\mathrm{rank}(\boldsymbol{AA}')=\mathrm{rank}(\boldsymbol{A}) \tag{11-34d}$$

如果 $\boldsymbol{A}$ 是 n 阶的方阵，则当且仅当 det($\boldsymbol{A}$)不等于零时，

$$\mathrm{rank}(\boldsymbol{A})=n$$

11. 矩阵的特征值和特征矢量

一个矩阵 $\boldsymbol{A}$，如果有

$$\boldsymbol{A}\boldsymbol{\gamma}_1=\lambda_1\boldsymbol{\gamma}_1;\cdots;\boldsymbol{A}\boldsymbol{\gamma}_k=\lambda_k\boldsymbol{\gamma}_k \tag{11-35}$$

则称满足上述方程的矢量 $\gamma_i(i=1,2,\cdots,k)$为矩阵 $\boldsymbol{A}$ 的相对于特征值 $\lambda_i(i=1,2,\cdots,k)$的特征矢量。如果矩阵 $\boldsymbol{A}$ 为对称矩阵，则所有特征值都是实数。

如果存在一非奇异矩阵 Γ，使得 $\boldsymbol{B}=\Gamma\boldsymbol{A}\Gamma^{-1}$，则方阵 $\boldsymbol{B}$ 和 $\boldsymbol{A}$ 被说成是相似的。如果 Γ 是正交矩阵，则称 $\boldsymbol{B}$ 是 $\boldsymbol{A}$ 的正交相似变换矩阵。

对于任一对称矩阵 $\boldsymbol{A}$，可经正交对称相似变换成为一对角矩阵 Λ，即

$$\Gamma'\boldsymbol{A}\Gamma=\Lambda \qquad 或 \qquad \boldsymbol{A}=\Gamma\Lambda\Gamma'$$

可以证明，其中的对角元素 $\lambda_i(i=1,2,\cdots,k)$就是矩阵 $\boldsymbol{A}$ 的特征值，而组成正交矩阵 Γ 的列矢量 $\gamma_i(i=1,2,\cdots,k)$则是其相对应的特征矢量，而且，矩阵 $\boldsymbol{A}$ 的秩就等于对角矩阵 Λ 中对角元素不为零的特征值的个数。

12. 广义特征值问题

这里所说的广义特征值问题，是指线性广义特征值问题

$$\boldsymbol{A}\gamma=\lambda\boldsymbol{B}\gamma \tag{11-36}$$

其中，$\boldsymbol{A}$，$\boldsymbol{B}$ 为 $n\times m$ 阶实数矩阵，γ 为 n 维非零列矢量，λ 为一数。由上述方程(11-36)表述的问题称为广义特征值问题，并称满足方程(11-36)的矢量 γ 为矩阵 $\boldsymbol{A}$ 相对于 $\boldsymbol{B}$ 的广义特征矢量，λ 为相应特征值。当 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 都为方阵时，可采用 QZ 算法求解，有关其具体解法可参阅文献[2]。

13. 幂等矩阵

一个方阵 $\boldsymbol{A}$，如果满足 $\boldsymbol{A}^2=\boldsymbol{AA}=\boldsymbol{A}$，则称为幂等阵。例如$\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$就是一个幂等阵。幂等阵具有下述性质：

① 幂等阵的特征值非 0 即 1。

证明：若 α 为幂等阵 $\boldsymbol{A}$ 的特征值，则有

$$\boldsymbol{Ab}=\alpha\boldsymbol{b} \tag{11-37}$$

式中，$\boldsymbol{b}$ 是非零矢量，它是 $\boldsymbol{A}$ 的属于特征值 α 的特征矢量，两边同左乘矩阵 $\boldsymbol{A}$，得

$$\boldsymbol{A}^2\boldsymbol{b}=\alpha\boldsymbol{Ab}=\alpha^2\boldsymbol{b} \tag{11-38}$$

同时，$\boldsymbol{A}$ 为幂等阵，即

$$Ab=\alpha b\Rightarrow A^2b=\alpha Ab=\alpha^2 b\Rightarrow \quad \alpha=\alpha^2 \tag{11-39}$$

方程式(11-39),立即可得 $\alpha=1$ 或 $\alpha=0$。

② 若矩阵 A 对称且幂等,则该矩阵的秩[rank(A)]等于该矩阵的迹。

证明:设对称幂等阵 A 的秩为 r,由于 A 是对称矩阵,故存在正交矩阵 Q,使得

$$A=Q\begin{bmatrix}\lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \ddots & & \\ & & & \lambda_{n-1} & \\ & & & & \lambda_n\end{bmatrix}Q^t=Q\Lambda Q^t \tag{11-40}$$

式中,$\lambda_1,\lambda_2,\cdots,\lambda_n$ 是 A 的特征值,A 又是秩为 r 的幂等阵,而幂等阵的特征值非 0 即 1,所以 λ_1,$\lambda_2,\cdots,\lambda_n$ 恰有 r 个 1,其余 $n-r$ 个全为零,于是

$$\mathrm{tr}(A)=\mathrm{tr}(Q\Lambda Q^t)=\mathrm{tr}(\Lambda Q^tQ)=\mathrm{tr}(\Lambda)=\sum\lambda_i=r=\mathrm{rank}(A)$$

14. 广义逆矩阵

一个 $n\times m$ 阶矩阵 A,若存在一个 $m\times n$ 阶矩阵 B,满足等式(11-41),

$$ABA=A \tag{11-41}$$

则称 B 是 A 的广义逆矩阵。通常记为 A^-,也称"－"逆。若 A 是一个满秩矩阵,由式(11-41)可看出,A^-就是 A^{-1}。因此广义逆确是通常逆矩阵的一种推广。问题是:对任何矩阵,广义逆是否一定存在?什么时候是唯一的?如果在式(11-41)的基础上再增加一些条件,使得所定义的广义逆存在并且唯一,这样的广义逆称"＋"逆或称 Moore-Penrose 逆,记为 A^+,它要求满足 4 个条件:

① $AA^+A=A$;

② $A^+AA^+=A^+$;

③ $(AA^+)^t=AA^+$;

④ $(A^+A)^t=A^+A$;

条件①和条件②称为反身性;条件③和条件④称为对称性。容易证明,Moore-Penrose 逆是存在且唯一的。Moore-Penrose 逆或称"＋"逆具有如下性质:

$$\mathrm{rank}(A)=\mathrm{rank}(A^+)$$
$$(A^+)^+=A$$
$$(A^t)^+=(A^+)^t$$
$$((AB)^+)^t=(A^tB^t)^+$$

AA^+ 和 A^+A 都为对称幂等阵。

15. 矩阵的导数

矩阵 A 对变量 t 的导数仍是一个矩阵,用 $\mathrm{d}A/\mathrm{d}t$ 表示,该导数矩阵的元素是矩阵 A 中相应元素对变量 t 求导的结果,若 a_{ij}为第 i 行第 j 列元素,则 $\mathrm{d}A/\mathrm{d}t$ 的第 i 行第 j 列的元素为 $\mathrm{d}a_{ij}/\mathrm{d}t$。

矩阵的导数有如下性质:

$$\mathrm{d}(AB)/\mathrm{d}t=(\mathrm{d}A/\mathrm{d}t)B+(\mathrm{d}B/\mathrm{d}t)A$$
$$\mathrm{d}[\mathrm{tr}(A)]/\mathrm{d}t=\mathrm{tr}(\mathrm{d}A/\mathrm{d}t)$$

16. 矢量函数的导数

设 f 是关于矢量 x 的数量函数,$f=f(x)$,则 f 关于矢量 x 的导数定义为

$$df(\boldsymbol{x})/d\boldsymbol{x}=[df/dx_1, df/dx_2, df/dx_3, \cdots, df/dx_n]^t$$

式中，$\boldsymbol{x}=[x_1, x_2, \cdots, x_n]^t$。矢量的导数有如下性质：

① 若 $\boldsymbol{a}$ 为常数矢量，则

$$d(\boldsymbol{a}^t\boldsymbol{x})/d\boldsymbol{x}=a$$

$$d(\boldsymbol{x}^t a)/d\boldsymbol{x}=a$$

② $d(\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x})/d\boldsymbol{x}=2\boldsymbol{A}\boldsymbol{x}$，这里 $\boldsymbol{A}$ 是对称方阵，且 $\boldsymbol{A}$ 中的各元素都是常数。

三、独立性、正交性和子空间

在 n 维实数空间中，如果

$$\sum \lambda_i \boldsymbol{a}_i=0 \qquad \Leftrightarrow \qquad \lambda_1=\lambda_2=\cdots=\lambda_n=0$$

那么，矢量组 $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ 是线性无关的。否则，若存在（$\lambda_1, \lambda_2, \cdots, \lambda_n$ 不全为零而使得 $\sum\lambda_i\boldsymbol{a}_i=0$ 成立，则称矢量组 $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ 是线性相关的。

在 n 维实数空间的子空间是一个子集，这个子集也是一个矢量空间。矢量组 $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ 的所有线性组合构成一个集合，被称为矢量组 $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ 张成的子空间，记为

$$\text{span}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}=\{\sum \lambda_i \boldsymbol{a}_i \mid \lambda_1, \lambda_2, \cdots, \lambda_n \in R\}$$

如果矢量组 $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ 是线性无关的，并且 $\boldsymbol{b}\in \text{span}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$，那么，$\boldsymbol{b}$ 是矢量组 $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ 的一个唯一的线性组合。

如果 $S_1, S_2, \cdots, S_n$ 是实数空间的子空间，那么它们的和也是一个子空间。子空间的交集也是一个子空间。

如果 $\{\boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \cdots, \boldsymbol{a}_{i_k}\}$ 是线性无关的，并且不是 $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ 的任一线性无关子集的真子集，这个子集 $\{\boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \cdots, \boldsymbol{a}_{i_k}\}$ 就是 $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ 的一个最大线性无关子集，那么

$$\text{span}\{\boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \cdots, \boldsymbol{a}_{i_k}\}=\text{span}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$$

并且 $\{\boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \cdots, \boldsymbol{a}_{i_k}\}$ 是 $\text{span}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$ 的一组基矢量。如果 S 是 n 维实数空间的一个子空间，那么在 S 中存在无关的基矢量 $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_k\}$，使得

$$S=\text{span}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_k\}$$

一个子空间 S 的任一组基矢量的个数必然相同。这个数就是该子空间的维数，记为 $\dim(S)$。

在实数空间中有两个与矩阵 $\boldsymbol{A}=[\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n]$ 有关的重要的子空间。$\boldsymbol{A}$ 的值域子空间定义为

$$R(\boldsymbol{A})=\{\boldsymbol{y}=\sum \lambda_i \boldsymbol{a}_i \mid \lambda_1, \lambda_2, \cdots, \lambda_n \in R\}=\text{span}\{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n\}$$

$\boldsymbol{A}$ 的零子空间定义为

$$N(\boldsymbol{A})=\{\boldsymbol{0}=\sum \mu_i \boldsymbol{a}_i \mid \mu_1, \mu_2, \cdots, \mu_n \in R\}$$

矩阵的秩定义为

$$\text{rank}(\boldsymbol{A})=\dim[R(\boldsymbol{A})]$$

可以证明 $\text{rank}(\boldsymbol{A})=\text{rank}(\boldsymbol{A}^t)$，于是一个矩阵的秩就等于其最大无关行数或最大无关列数。

对于任一矩阵 $\boldsymbol{A}_{n\times m}$ $(n\leqslant m)$，则

$$\dim[N(\boldsymbol{A})]+\dim[R(\boldsymbol{A})]=n$$

如果 $\boldsymbol{A}$ 为一方阵，那么下面的说法都是等价的：

① $\boldsymbol{A}$ 是非奇异的

② $\dim[N(\boldsymbol{A})]=0$

③ $\operatorname{rank}(\boldsymbol{A})=n$

在 n 维实数空间中，如果矢量组$\{\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n\}$满足 $i\neq j$，就有 $\boldsymbol{a}_i^t\boldsymbol{a}_j=0$，则称此矢量组是正交的，如果满足 $\boldsymbol{a}_i^t\boldsymbol{a}_j=\delta_{ij}(\delta_{ij}=0$，当 $i\neq j$；$\delta_{ij}=1$，当 $i=j)$，则称此矢量组是标准正交的。更一般地，在 n 维实数空间中的子空间$\{S_1,S_2,\cdots,S_n\}$称为相互正交，是指如果 $\boldsymbol{x}\in S_i$ 和 $\boldsymbol{y}\in S_j$，并且 $i\neq j$就有

$$\boldsymbol{x}^t\boldsymbol{y}=0$$

一个 n 维子空间 S 的正交补定义为

$$S^{\perp}=\{\boldsymbol{y}\in n\text{ 维空间 } \boldsymbol{x}^t\boldsymbol{y}=0,\boldsymbol{x}\in S\}$$

如果矢量组$\{\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_k\}$是标准正交的并张成 n 维子空间 S，则称矢量组$\{\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_k\}$构成子空间的一个标准正交基。它总可以扩充为 n 维空间的一组完全的标准正交基$\{\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_k,\cdots,\boldsymbol{a}_n\}$。此时

$$S^{\perp}=\operatorname{span}\{\boldsymbol{a}_{k+1},\cdots,\boldsymbol{a}_n\}$$

四、矢量范数和矩阵范数

在化学计量学的算法中，经常要求我们评价矢量或矩阵的大小，对此矢量范数和矩阵范数是非常有用的。

1．矢量范数

在 n 维实数空间的矢量 $\boldsymbol{x}$ 的范数是一个这样的函数 $f(\boldsymbol{x})$，它具有下列性质：

$f(\boldsymbol{x})\geqslant 0$　　对所有的 n 维实数空间矢量 $\boldsymbol{x}$，等号当且仅当 $\boldsymbol{x}=0$ 时成立；

$f(\boldsymbol{x}+\boldsymbol{y})\leqslant f(\boldsymbol{x})+f(\boldsymbol{y})$，对所有的 n 维实数空间矢量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 成立；

$f(a\boldsymbol{x})=|a|f(\boldsymbol{x})$　对所有的 n 维实数空间矢量 $\boldsymbol{x}$ 成立。

我们用 $\|\boldsymbol{x}\|$ 表示该函数 $f(\boldsymbol{x})$，在双竖线下加下标来区别不同的范数。

一种有用的范数是 Holder 范数或称 p 范数，其定义为

$$\|\boldsymbol{x}\|_p=(|x_1|^p+|x_2|^p+\cdots+|x_n|^p)^{1/p} \tag{11-42}$$

式中

$$\|\boldsymbol{x}\|_1=(|x_1|+|x_2|+\cdots+|x_n|) \qquad \text{1-范数}$$

$$\|\boldsymbol{x}\|_2=(|x_1|^2+|x_2|^2+\cdots+|x_n|^2)^{1/2} \qquad \text{2-范数}$$

$$\|\boldsymbol{x}\|_\infty=\max(|x_i|) \qquad \text{极大范数}$$

这 3 个范数是最重要的范数。关于 2-范数的一个古典结论是 Holder 不等式

$$|\boldsymbol{x}^t\boldsymbol{y}|\leqslant\|\boldsymbol{x}\|_p\|\boldsymbol{y}\|_q \qquad 1/p+1/q=1$$

它的一个非常重要的特殊情况是 Cauchy-Schwatz 不等式：

$$|\boldsymbol{x}^t\boldsymbol{y}|\leqslant\|\boldsymbol{x}\|_2\|\boldsymbol{y}\|_2$$

其中，在正交变换下 2-范数是不变的，因为如果 $\boldsymbol{Q}^t\boldsymbol{Q}=\boldsymbol{I}$，则

$$\|\boldsymbol{Q}\boldsymbol{x}\|_2^2=\boldsymbol{x}^t\boldsymbol{Q}^t\boldsymbol{Q}\boldsymbol{x}=\boldsymbol{x}^t\boldsymbol{x}=\|\boldsymbol{x}\|_2^2$$

2．矩阵范数

矩阵范数的定义与矢量范数的定义应当是等价的，在实数空间的矩阵 $\boldsymbol{A}$ 的范数 $f(\boldsymbol{A})$也具有下列性质：

$f(\boldsymbol{A})\geqslant 0$　　对所有的实数空间矩阵 $\boldsymbol{A}$，等号当且仅当 $\boldsymbol{A}=0$ 时成立；

$f(\boldsymbol{A}+\boldsymbol{B})\leqslant f(\boldsymbol{A})+f(\boldsymbol{B})$　对所有的实数空间矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 成立；

$f(a\boldsymbol{A})=|a|f(\boldsymbol{A})$　对所有的实数空间矩阵 $\boldsymbol{A}$ 成立。

用 $\|\boldsymbol{A}\|$ 表示该函数 $f(\boldsymbol{A})$，与矢量范数一样，矩阵范数也用加双竖线来表示。

在数值分析中，最常用的范数是 Frobenius 范数或称 F 范数，其定义为

$$\|\boldsymbol{A}\|_F=(\sum\sum|a_{ij}|^2)^{1/2} \tag{11-43}$$

和 p 范数

$$\|\boldsymbol{A}\|_p=\sup(\|\boldsymbol{A}\boldsymbol{x}\|_p/\|\boldsymbol{x}\|_p) \tag{11-44}$$

在本定义中用的 p 范数就是矢量范数已定义过的 p 范数。在此定义的矩阵的 p 范数也是一族范数。矩阵 p 范数具有下述性质：

$$\|\boldsymbol{AB}\|_p\leqslant\|\boldsymbol{A}\|_p\|\boldsymbol{B}\|_q$$

$$\|\boldsymbol{A}\boldsymbol{x}\|_p\leqslant\|\boldsymbol{A}\|_p\|\boldsymbol{x}\|_q$$

特别地，矩阵的 Frobenius 范数和 2 范数（矩阵 p 范数的一种）在正交变换下是不变的，对所有的维数合适的正交矩阵 $\boldsymbol{Q}$ 和 $\boldsymbol{P}$ 有

$$\|\boldsymbol{QAP}\|_2=\|\boldsymbol{A}\|_2$$

$$\|\boldsymbol{QAP}\|_F=\|\boldsymbol{A}\|_F$$

五、张　量

已知两个矢量的外积产生一个矩阵，即

$$\boldsymbol{ab}^t=\begin{bmatrix}a_1\\a_2\\\vdots\\a_n\end{bmatrix}[b_1,b_2,\cdots,b_m]=\begin{bmatrix}a_1b_1 & a_1b_2 & \cdots & a_1b_m\\a_2b_1 & a_2b_2 & \cdots & a_2b_m\\\vdots & \vdots & & \vdots\\a_nb_1 & a_nb_2 & \cdots & a_nb_m\end{bmatrix}$$

然而，对于三个矢量的外积，用上述方法就很难表出了。为此，定义 $\boldsymbol{a}\otimes\boldsymbol{b}\otimes\boldsymbol{c}$ 为 $\boldsymbol{a}$、$\boldsymbol{b}$、$\boldsymbol{c}$ 这三个矢量的外积，它的结果将产生一个形如立体的数据，在此记为 $\boldsymbol{G}$，即

$$\boldsymbol{G}=\boldsymbol{a}\otimes\boldsymbol{b}\otimes\boldsymbol{c}=(G_{ijk})(i=1,2,\cdots,n;j=1,2,\cdots,m;k=1,2,\cdots,d)$$

这样由三个矢量的外积而产生的张量，一般称为三线性张量，这种三线性张量在多元分辨中有着特殊重要的地位。在化学计量学中，一般都是把这样的三维张量展开成矩阵来计算。

参 考 文 献

1　张尧庭，方开泰. 多元统计分析引论. 北京：科学出版社，1982

2　Golub G H, Van Loan C F. Matrix Coputation. The Johns Hopkins University Press, 1983

3　Magnus J. R, Neudecker H. Matrix differential Calculus with Applications in Statistics and Econometrics. John Wiley & Sons, 1990

第二篇 MATLAB 原程序及其应用

第十二章 使用 MATLAB 语言进行化学计量学研究编程运算简介

化学计量学需大量地使用数学、统计学和计算机科学知识，编制计算机程序是进行化学计量学研究的必要基础。然而，用经典高级语言编制程序是繁杂的，消耗大量人力且工作进度与效率不高。随着计算机科学的飞速发展，编制程序的方法也在长足进步，由美国 Mathwork 公司推出的 MATLAB 是一种功能强、效率高便于进行科学和工程计算的交互式软件包。它提供了一种全新的编程语言，其语法规则简单，很贴近数学运算思维方式。用 MATLAB 写程序，犹如在一张演算纸上排列公式和求解问题，编程效率很高，因此称为"演算纸式的"科学工程的算法语言。

由于使用的化学计量学程序大多数都需涉及大量的矩阵运算，如采用一般高级语言，如 BASIC、FORTRAN、PASCAL 和 C 语言，来进行矩阵运算，不但需要很多子程序，而且有些较新的算法，如奇异值分解、广义特征值问题的 QZ 算法、人工神经网络、小波分析等，还很难找到这些高级语言的相应的子程序。这就使化学计量学研究成果的普及受到一定程度的限制。而 MATLAB 语言则正是在矩阵运算的方面，显得特别简捷、高效和方便。在化学计量学的研究和应用中，由于大都是进行多变量运算，矢量和矩阵是基本运算单元，所以，MATLAB 语言很快就成为其研究的主要工具，目前国际上两本化学计量学专业杂志，"Journal of Chemometris"和"Chemometrics and Intelligent Laboratory Systems"多次对其进行了专门的介绍，使其几乎成为进行化学计量学研究的特定计算机编程语言，很多化学计量学的新算法在发表时，还都附上了 MATLAB 语言所编制的程序。要想在我国普及化学计量学，掌握使用 MATLAB 语言似很有必要。MATLAB 语言是一种类似于一般高级语言编程的更高级语言，它不但保持了一般高级语言特征，还特别地注意到了人机交互信息的优点（从这一角度来看，它与 BASIC 语言很类似），可将一个复杂编程来分步进行调试，特别适合进行新的算法研究。而且，随着 MATLAB 版本的不断更新，其功能越来越强，它所附带的一些工具箱(Tool Box)，使之在诸如一般数值计算、多元统计分析、数字信号处理、系统识别、自动控制、时序分析与建模、优化设计、人工神经网络、动态仿真系统、小波分析、特殊函数等领域，表现出一般计算机高级语言难以比拟的优势。特别值得提出的是，MATLAB 语言还直接提供了强有力的绘图指令，如绘制三维图，等高线图等，可使一些在一般高级语言中难以实现的绘图功能，在 MATLAB 语言中，一个指令就可完成。为化学计量学工作者提供了极大的方便，可使我们摆脱一些在计算机编程方面的技术性操作，而将主要精力集中到化学计量学新算法的思路研究之上。

随着计算机科学的不断普及，有关 MATLAB 语言的书籍也在相继出现（见第一章文献[4]）。在这些书籍中，主要介绍 MATLAB 语言的一些基本知识和概念，读者参看这些书籍，即可方便地知道 MATLAB 系统要素，MATLAB 语言的变量与语句，数值输入与输出格式，

MATLAB 系统工作空间信息，以及 MATLAB 的在线帮助功能等。对于这些知识我们在此就不作详细介绍了，读者可直接阅读这些书籍。其实，学习 MATLAB 语言，如果读者已有一般高级语言的基础，很短的时间就可完全进入角色，至于一些细节，读者可在继续的实践中不断提高。在这里，我们给出的就是一些让读者可直接进入 MATLAB 编制化学计量学一般程序一些最基本知识，希望读者经过阅读本节，对怎样使用 MATLAB 语言来进行化学计量学研究有个大致了解。有关 MATLAB 语言的一些细节问题，希望读者参阅有关 MATLAB 语言介绍的专门书籍。

一、MATLAB 系统简介

MATLAB 系统是由美国 Mathworks 公司推出的用于数值计算和图形处理的科学计算系统环境。MATLAB 是英文 Matrix Laboratory（矩阵实验室）缩写。它集中了日常数学处理中的各种功能，包括高效的数值计算、矩阵运算、信息处理和图形生成等功能。在 MATLAB 环境下，用户可以集成地进行程序设计、数值计算、图形绘制、输入输出、文件管理等各项操作。

MATLAB 提供了一个人机交互的数学系统环境，该系统的基本数据结构是矩阵，在构成矩阵对象时，不要求明确的维数说明。与利用 C 语言或 FORTRAN 语言作数值计算的程序设计相比，利用 MATLAB 可以节省大量的编程时间。在美国的一些大学里，MATLAB，正在成为对数值线性代数以及其他一些高等应用数学课程进行辅助教学的有益工具。在化学计量学研究中，MATLAB 是一种很好的算法研究工具。

MTALAB 系统由五个主要部分组成，下面分别加以介绍。

(1) MATLAB 语言体系　MATLAB 是高层次的矩阵/数组语言，具有条件控制、函数调用、数据结构、输入输出、面向对象等程序语言特性。利用它既可以进行小规模编程，完成算法设计和算法实验的基本任务，也可以进行大规模编程，开发复杂的应用程序。

(2) MATLAB 工作环境　MATLAB 工作环境是 MATLAB 提供给用户使用的管理功能的总称，包括管理工作空间中的变量，数据输入输出的方式和方法，以及开发、调试、管理文件的各种工具。

(3) 图形句柄 (handle) 系统　图形句柄 (handle) 系统是 MATLAB 图形系统的基础，包括完成数据图示、图像处理、动画生成、图形显示等功能的高层 MATLAB 命令，也包括用户对图形图像等对象进行特性控制的低层 MATLAB 命令，以及开发应用程序的各种工具。

(4) MATLAB数学函数库　这是对 MATLAB 使用的各种数学算法的总称，包括各种初等函数的算法，也包括矩阵运算、矩阵分析等高层次数学算法。

(5) MATLAB应用程序接口 (API)　MATLAB 为用户提供了一个函数库，使得用户能够在 MATLAB 环境中使用 C 程序或 FORTRAN 程序，包括从 MATLAB 中调用子程序（动态链接），读写 MAT 文件的功能。

综上所述，可以看出 MATLAB 是一个功能十分强大的系统，是集数值计算、图形管理、程序开发为一体的环境。除此之外，MATLAB 还具有很强的功能扩展能力，与它的主系统一起，可以配备各种各样的工具箱，以完成一些特定的任务。目前，MathWorks 公司推出了18 种工具箱。用户还可以根据自己的工作任务，开发自己的工具箱。

二、MATLAB 的工作环境

启动 MATLAB 系统后，Windows 的工作平台上会弹出一个窗口，命令窗口是用户与

MATLAB 解释器进行通信的工作环境，提示符“》”表示 MATLAB 解释器正等待用户输入命令。所有的 MATLAB 命令、MATLAB 函数，以及 MATLAB 程序都要在这个窗口下运行。

在命令窗口中，用户可以发出 MATLAB 命令。例如，为了生成一个 3×3 阶的矩阵，可在提示符下，键入如下的命令：

A=[1 2 3 ; 4 5 6 ; 7 8 9]

方括号命令表示矩阵，空格将每个的元素分开，而分号将矩阵的各行数值分开。再键入 Enter（回车）后，MATLAB 将回显如下的矩阵：

```
A=
1    2    3
4    5    6
7    8    9
```

为了求该矩阵的逆矩阵，则只要键入命令

》B=inv(A);

则可得到矩阵 A 的逆，其结果将存入矩阵 B 之中。如果不想在命令窗口中显示计算结果，只要如上所示，在该命令后多键入一个分号即可。此时，MATLAB 系统只完成该命令所要求的计算任务，其计算结果不回显。这项功能在程序设计中是非常有用的。

由于 MATLAB 的运算功能很强，所以，它的这种类似于 BASIC 语言的、具有很强的对话功能的工作环境对化学计量学程序调试和编制有很大好处。如计算最小二乘回归，输入了校正矩阵 X 和响应矢量 Y 后，在 MATLAB 的工作环境下，直接用以下语句就可得到需求解的参数估计矢量 C 来：

》C=inv (X′ * X) * X′ * Y

上述语句中，′表示矩阵的转置，* 表示乘号。为解释怎样来利用 MATLAB 进行化学计量学运算，我们在此先对 MATLAB 的一般运算符号和矩阵运算功能给出一般性介绍。

三、MATLAB 的一般运算符号和矩阵运算功能

在化学计量学的研究和应用中，由于大都是进行多变量运算，矢量和矩阵是基本运算单元，所以，熟悉 MATLAB 的矩阵运算功能是用 MATLAB 语言来进行化学计量学研究的基础。

在 MATLAB 系统中，只管理着一种对象（Object）——矩阵（包括复矩阵）。1×1 的矩阵实际上是一个标量即一个数量，任何数在 MATLAB 系统中是作为 1×1 的矩阵来处理的。与数学术语一样，仅有一行或一列的矩阵称为向量。MATLAB 的大部分运算或命令是在矩阵运算的意义下执行的。

（一）MATLAB 的基本运算符

1. 矩阵（算术）运算符

MATLAB 的主要矩阵（算术）运算符如下所列：

A′：表示矩阵 A 的转置，如果 A 是复矩阵，则其运算的结果是共轭转置；

A+B：表示矩阵 A 和 B 的和，表示两个矩阵的元素对应相加．如果 A 和 B 都是标量，则表示两个数相加；

A−B：表示矩阵 A 和 B 的差，表示两个矩阵的元素对应相减．如果 A 和 B 都是标量，则表示两个数相减；

A * B：表示矩阵 A 和 B 的乘法，A 与 B 均可以是向量或标量，只要符合矩阵乘法的定义；

A. * B：表示矩阵A和B的对应元素相乘，A和B为同维数的矩阵，除非其中之一为标量；

A/B：表示方程 $x*A=B$ 的解 x；

A\B：表示方程 $A*x=B$ 的解 x；

A./B：表示矩阵 A 的元素除以矩阵 B 的对应元素，即等于[$A(i,j)/B(i,j)$]，A 和 B 为同维数的矩阵，除非其中之一为标量；

A.\B：表示矩阵 B 的元素除以矩阵 A 的对应元素，即等于[$B(i,j)/A(i,j)$]，A 和 B 为同维数的矩阵，除非其中之一为标量。

2. 矩阵的创建

可以使用下列任何一种方法在 MATLAB 环境下创建或输入一个矩阵：

① 显示地输入一个元素序列；

② 用 MATLAB 的内部函数创建一个矩阵；

③ 在 M 文件中用 MATLAB 语句创建一个矩阵；

④ 从一个外部数据文件中装载并创建一个矩阵。

在 MATLAB 环境中，不需要对创建的变量对象给出类型说明和维数说明，所有的变量都作为双精度的矩阵来分配内存空间和存储空间。MATLAB 将自动地为每一个变量分配内存。最简单的创建矩阵的方法是显示地输入矩阵的元素序列。具体方法如下：将矩阵的元素用方括号括起来，按矩阵行的顺序输入各元素，元素与元素之间用空格或逗号分开，用分号将每行的元素分开。例如，在键入下列的 MATLAB 语句：

```
》 A=[1  3  5  7 ; 1  2  3  5  3 ; 3  5  9  1]
```

后，MATLAB 执行该语句的输出结果是

```
A=
     1    3    5    7
    12    3    5    3
     3    5    9    1
```

或者，键入

```
》a=[2,3,4,6;1,4,5,7]
```

它的输出结果是：

```
a=
   2    3    4    6
   1    4    5    7
```

注意，行与行之间都是用分号来隔开的。

3. MATLAB 的常用数学函数

MATLAB 提供了几乎所有的常用的初等函数，函数的变量在 MATLAB 中被规定为矩阵变量，运算法则是将函数逐项作用于矩阵的元素上，因而运算的结果是一个与自变量同维数的矩阵。例如：

```
》A=[123 245 365
345 345 232]
》B=fix(0.45*A)
》C=cos(A)
```

这三条语句运算的结果分别是

```
A=
    123   245   365
    345   345   232
B=
     55   110   164
    155   155   104
C=
   -0.8880    0.9990    0.8391
    0.8391    0.8391    0.8880
```

MATLAB 提供的三角函数主要包括有

sin　　正弦函数；
cos　　余弦函数；
tan　　正切函数；
asin　　反正弦函数；
acos　　反余弦函数；
atan　　反正切函数；
sinh　　双曲正弦函数；
cosh　　双曲余弦函数；
tanh　　双曲正切函数；
asinh　　反双曲正弦函数；
acosh　　反双曲余弦函数；
atanh　　反双曲正切函数。

MATLAB 提供的初等函数包括有：

abs　　实数的绝对值、复数的模、字符串的 ASCⅡ码值；
angle　　复数的幅角；
sqrt　　方根函数；
real　　复数的实部；
imag　　复数的虚部；
conj　　复共轭运算；
round　　最邻近整数截断（四舍五入）；
fix　　向零方向截断为整数；
floor　　不大于自变量的最大整数；
ceil　　不小于自变量的最小整数；
sign　　符号函数；
rem　　求余数或模运算；
gcd　　最大公因子；
lcm　　最小公倍数；
exp　　自然指数函数（以 e 为底）；
log　　自然对数函数（以 e 为底）；

log10　　以 10 为底的对数函数。

为了方便应用，MATLAB 还提供了一些特殊的数学函数，主要包括有：

bessel　　Bessel 函数；

beta　　完全与不完全 Beta 函数；

rat　　有理逼近；

erf　　误差函数；

erfinv　　逆误差函数。

上述函数基本上都是按矩阵的元素进行运算。

4. 几个可方便产生矩阵和矢量的方法

直接从 MATLAB 的命令窗口输入矩阵和矢量确实是方便，但是，这样做的效率不高。在 MATLAB 中，有几个可方便产生矩阵和矢量的方法，这将使我们的编程效率提高，在此我们给出简要介绍。

冒号运算符使得数据向量的创建变得十分容易。为了得到一个竖型的函数数据表，只需先用冒号运算符生成行向量，再转置，计算出函数值列，就可用两列数据拼成一个函数数据表，例如：

```
》 x=[0.0:0.2:3.0]';
》 y=-exp(x).*sin(x);
》 [x y]
ans=
        0            0
   0.2000      -0.2427
   0.4000      -0.5809
   0.6000      -1.0288
   0.8000      -1.5965
   1.0000      -2.2874
   1.2000      -3.0945
   1.4000      -3.9962
   1.6000      -4.9509
   1.8000      -5.8914
   2.0000      -6.7188
   2.2000      -7.2967
   2.4000      -7.4457
   2.6000      -6.9406
   2.8000      -5.5088
   3.0000      -2.8345
```

能够创建向量的其他函数还有：logspace 函数，用于创建对数等距的向量；linspace 函数，用于创建指定长度的等距向量。例如：

》LINSPACE($x1$, $x2$, n)就可产生一个含 n 个元素的行向量，它的第一个元素是 $x1$，最后一个元素是 $x2$，元素与元素之间的差值是相等的。

如键入：

》linspace(.3,5.2,14)

所得结果显示如下：

```
ans =
    Columns 1 through 7
      0.3000    0.6769    1.0538    1.4308    1.8077    2.1846    2.5615
    Columns 8 through 14
      2.9385    3.3154    3.6923    4.0692    4.4462    4.8231    5.2000
```

如果将 LINSPACE($x1$,$x2$,n)中的 n 略去，如 LINSPACE($x1$,$x2$)，MATLAB 就产生一个含 100 个元素的行向量。

MATLAB 还提供了一类函数，可用于创建某些特殊类型矩阵，这些函数主要包括有：

diag	向量的对角矩阵；
hadamard	Hadamard 矩阵；
magic	幻方矩阵；
pascal	杨辉三角形矩阵；
zeros	元素全为零的矩阵；
ones	元素全为 1 的矩阵；
rand	元素为均匀分布的随机矩阵；
randn	元素为正态分布的随机矩阵；
eye	单位矩阵。

例如，为了创建一个 3×5 的随机矩阵，只需输入下列语句：

》rand(3,5)

即可得

```
ans=
     0.9501  0.4860  0.4565  0.4447  0.9218
     0.2311  0.8913  0.0185  0.6154  0.7382
     0.6068  0.7621  0.8214  0.7919  0.1763
```

这为我们进行计算机数字模拟带来很大方便。

5. MATLAB 的矩阵角标系统

为标记矩阵的每个元素的位置，数学上用该元素在矩阵中的行、列标号来表示。MATLAB 也使用类似的方法来表示，这与一般计算机高级语言没什么差别。不同的是，MATLAB 约定矩阵的角标可以是一个向量。对矩阵来说，可以利用向量角标访问矩阵的子矩阵就十分方便了。MATLAB 的矩阵角标系统在化学计量学的应用中起到十分重要的作用。

例如，设 A 是一个 10×10 的矩阵，那么语句

》A(1：5，3)

表示创建一个 5×1 的列向量，由矩阵 A 的第三列前 5 个元素组成，又如语句

》A(1：5，7：10)

表示创建一个新的 5×4 阶矩阵，由矩阵 A 的前五行后四列的元素组成，即为 A 的一个子块。在这种表示方法中，单用冒号作角标时，表示该矩阵的全部行或全部列。例如，语句

》A(:，3)

表示矩阵 A 的第三列，而

```
》A(1:5,:)
```

表示矩阵 A 的前五行。

矩阵的角标表示可以用在赋值语句中，这是MATLAB中矩阵运算的一种技巧。例如，键入

```
》B=magic(8)
```

得

```
B=
    64    2    3   61   60    6    7   57
     9   55   54   12   13   51   50   16
    17   47   46   20   21   43   42   24
    40   26   27   37   36   30   31   33
    32   34   35   29   28   38   39   25
    41   23   22   44   45   19   18   48
    49   15   14   52   53   11   10   56
     8   58   59    5    4   62   63    1
```

再键入

```
》a=rand(8)
```

得

```
a=
    0.9501  0.8214  0.9355  0.1389  0.4451  0.8381  0.3046  0.3784
    0.2311  0.4447  0.9169  0.2028  0.9318  0.0196  0.1897  0.8600
    0.6068  0.6154  0.4103  0.1987  0.4660  0.6813  0.1934  0.8537
    0.4860  0.7919  0.8936  0.6038  0.4186  0.3795  0.6822  0.5936
    0.8913  0.9218  0.0579  0.2722  0.8462  0.8318  0.3028  0.4966
    0.7621  0.7382  0.3529  0.1988  0.5252  0.5028  0.5417  0.8998
    0.4565  0.1763  0.8132  0.0153  0.2026  0.7095  0.1509  0.8216
    0.0185  0.4057  0.0099  0.7468  0.6721  0.4289  0.6979  0.6449
```

再键入

```
》a(:,[3 5 7])=B(:,1:3)
```

则可得

```
a=
    0.9501  0.8214  64.0000  0.1389   2.0000  0.8381   3.0000  0.3784
    0.2311  0.4447   9.0000  0.2028  55.0000  0.0196  54.0000  0.8600
    0.6068  0.6154  17.0000  0.1987  47.0000  0.6813  46.0000  0.8537
    0.4860  0.7919  40.0000  0.6038  26.0000  0.3795  27.0000  0.5936
    0.8913  0.9218  32.0000  0.2722  34.0000  0.8318  35.0000  0.4966
    0.7621  0.7382  41.0000  0.1988  23.0000  0.5028  22.0000  0.8998
    0.4565  0.1763  49.0000  0.0153  15.0000  0.7095  14.0000  0.8216
    0.0185  0.4057   8.0000  0.7468  58.0000  0.4289  59.0000  0.6449
```

从上述过程可以看出，矩阵 a 的第3列、第5列和第7列分别被矩阵 B 的前三列代替了。

一般地，如果 v 和 w 都是整数元素向量，那么 $A(v,w)$ 表示的是一个取自 A 的矩阵子块，其行由向量 v 确定，而列由向量 w 确定，称 v 为行角标、w 为列角标。

另外，在 MATLAB 中，用这样的方法可得到矩阵 a 的列反序的矩阵，如键人

```
》a(:,8:-1:1)
```

就可得

```
ans=
    0.3784   3.0000  0.8381   2.0000  0.1389  64.0000  0.8214  0.9501
    0.8600  54.0000  0.0196  55.0000  0.2028   9.0000  0.4447  0.2311
    0.8537  46.0000  0.6813  47.0000  0.1987  17.0000  0.6154  0.6068
    0.5936  27.0000  0.3795  26.0000  0.6038  40.0000  0.7919  0.4860
    0.4966  35.0000  0.8318  34.0000  0.2722  32.0000  0.9218  0.8913
    0.8998  22.0000  0.5028  23.0000  0.1988  41.0000  0.7382  0.7621
    0.8216  14.0000  0.7095  15.0000  0.0153  49.0000  0.1763  0.4565
    0.6449  59.0000  0.4289  58.0000  0.7468   8.0000  0.4057  0.0185
```

这样，MATLAB 用来进行矩阵的分块，重排等很难在一般计算机高级语言实现的功能变得十分容易了。

在化学计量学中，我们经常需要将矩阵进行矢量化而后计算，这在 MATLAB 中也是很容易实现的，例如，语句

```
》A=[1  2; 3  4; 5  6]
》b=A(:)
```

的输出结果是

```
A:  =
1    2
3    4
5    6
b:  =
1
2
3
4
5
6
```

值得提出的是，$A(:)$语句如在赋值语句的左端，还可表示用一个向量对矩阵 A 来进行顺序赋值，只是此时矩阵 A 必须事先存在。例如，设 A 是上述的 3×2 阶的矩阵，那么语句

```
》A(:)=11:16
```

则表示行向量 [11，12，13，14，15，16] 的 6 个元素依照矩阵 A 的列顺序给 A 的元素赋值，但保持矩阵 A 的维数 3×2，因此，

```
A=
11    14
12    15
13    16
```

MATLAB 的函数 reshape 正是使用这种方法来改变一个矩阵的维数的。一般说来，矩阵的角标出现在语句等号的右端时，表示创建一个新的矩阵对象；在左端时，则表示对原矩阵中的部分或全部元素重新赋值。

在 MATLAB 中，还可用所谓的矩阵的 0-1 角标来定义矩阵子块，这样就可以用关系运算的结果来引用矩阵的元素或子块，这在一些数学计算中是十分有意义的。假设 a=rand(8)是一个 8×8 阶随机数矩阵，L 是一个长度为 m 的 0-1 向量，那么 $A(L, I)$ 表示由向量 L 的非零元素对应的矩阵 A 的行组成的子块。如，

```
a=rand(8)
```

得

```
》a=
    0.4387  0.7446  0.1338  0.0272  0.6085  0.7176  0.6992  0.2731
    0.4983  0.2679  0.2071  0.3127  0.0158  0.6927  0.7275  0.2548
    0.2140  0.4399  0.6072  0.0129  0.0164  0.0841  0.4784  0.8656
    0.6435  0.9334  0.6299  0.3840  0.1901  0.4544  0.5548  0.2324
    0.3200  0.6833  0.3705  0.6831  0.5869  0.4418  0.1210  0.8049
    0.9601  0.2126  0.5751  0.0928  0.0576  0.3533  0.4508  0.9084
    0.7266  0.8392  0.4514  0.0353  0.3676  0.1536  0.7159  0.2319
    0.4120  0.6288  0.0439  0.6124  0.6315  0.6756  0.8928  0.2393
```

则如键入

```
》L=a(:,3)>=.5
```

可得

```
L=
    0
    0
    1
    1
    0
    1
    0
    0
```

再键入

```
》b=a(L,:)
```

可得到 a 矩阵的一个子块：

```
b=
    0.2140  0.4399  0.6072  0.0129  0.0164  0.0841  0.4784  0.8656
    0.6435  0.9334  0.6299  0.3840  0.1901  0.4544  0.5548  0.2324
    0.9601  0.2126  0.5751  0.0928  0.0576  0.3533  0.4508  0.9084
```

下面的语句则可以用来直接消去矩阵 b 中大于 0.3 的元素，而且还可将其矢量化：

```
》c=b(b<=.3)
```

```
c=
    0.2140
    0.2126
    0.0129
    0.0928
    0.0164
    0.1901
    0.0576
    0.0841
    0.2324
```

同样，MATLAB 可以由小矩阵作为矩阵子块来构造大矩阵。例如，如果 A 是 n 阶方阵，那么，语句

```
》C=[AA'; ones(size(A))A.^2];
```

表示创建一个 $2n$ 阶的方阵。在这样的运算中，要保持矩阵阶数的协调性，否则 MATLAB 系统会报告运算出错。

示例：如键入

```
》x=magic(4)
```

得

```
x=
   16    2    3   13
    5   11   10    8
    9    7    6   12
    4   14   15    1
```

再键入

```
》y=[x x';ones(size(x))x.^.5]
```

则可得

```
y=
   16.0000   2.0000   3.0000  13.0000  16.0000   5.0000   9.0000   4.0000
    5.0000  11.0000  10.0000   8.0000   2.0000  11.0000   7.0000  14.0000
    9.0000   7.0000   6.0000  12.0000   3.0000  10.0000   6.0000  15.0000
    4.0000  14.0000  15.0000   1.0000  13.0000   8.0000  12.0000   1.0000
    1.0000   1.0000   1.0000   1.0000   4.0000   1.4142   1.7321   3.6056
    1.0000   1.0000   1.0000   1.0000   2.2361   3.3166   3.1623   2.8284
    1.0000   1.0000   1.0000   1.0000   3.0000   2.6458   2.4495   3.4641
    1.0000   1.0000   1.0000   1.0000   2.0000   3.7417   3.8730   1.0000
```

为了实现由一个矩阵创建新的矩阵，以及对矩阵的运算进行管理，MATLAB 提供了一些功能函数，主要包括有

rot90　将矩阵旋转 90 度；

fliplr　将矩阵的列反序（左右反序）；

flipud　将矩阵的行反序（上下反序）；

diag　　提取矩阵的对角元素形成对角矩阵或将一个向量转换成对角矩阵；

tril　　提取矩阵的下三角部分；

triu　　提取矩阵的上三角部分；

reshape　改变矩阵阶数。

例如，当要将一个3×4阶矩阵按照列的顺序重排成2×6的矩阵时，可以用下列的语句来实现

```
》A=[1 4 7 10;2 5 8 11;3 6 9 12]
```

得

```
A=
    1    4    7    10
    2    5    8    11
    3    6    9    12
```

再键入

```
》B=reshape(A,2,6)
```

得

```
B=
1    3    5    7    9     11
2    4    6    8    10    12
```

又如，A 的下三角矩阵可以由下列语句得到

```
》tril(A)
```

得

```
ans=
   1  0  0  0
   2  5  0  0
   3  6  9  0
```

又如

```
》flipud(A)
```

得

```
ans=
      3  6  9  12
      2  5  8  11
      1  4  7  10
```

与矩阵运算相关的另外两个MATLAB函数是size和length。函数size(A)的返回值是一个二元素的向量，第一元素为矩阵 A 的行数，第二元素为矩阵 A 的列数。在MATLAB中，size的定义已经扩展到任何多维数组，它的返回值是一个维数向量。如果已经知道 v 是一个向量，那么length(v)的返回值是该向量的长度，或max(size(v))。

更重要的是，MATLAB还定义了一个很特殊的矩阵，即空矩阵。空矩阵可以由下列的语句创建：

```
》x=[ ]
```

即将一个0×0阶的矩阵赋给变量 x，随后即可以应用这个空矩阵变量，而不会导致错误。利

用空矩阵的特性，可以方便地进行从一个矩阵中消去部分行和部分列的运算。例如，语句

》A(:,[2,4])=[ ]

的运算结果是从矩阵 A 中消去第 2 列和第 4 列的矩阵子块。在 MATLAB 中，某些 MATLAB 函数用空矩阵作变量时，被赋予了特定的值。这些函数包括 det（矩阵行列式的值）、cond（矩阵的条件数）、prod（各列元素之积）和 sum（各列元素之和）。例如，如果 x 是一个空矩阵，则分别规定此 det(x)=1，prod(x)=1 和 sum(x)=0 等。注意，空矩阵在化学计量学的 MATLAB 编程中将具有十分重要的作用。

（二）矩阵与数组运算

为了方便数据分析，MATLAB 提供了一部分面向矩阵的列运算的数据分析函数，这些函数主要包括有

max	矩阵各列的最大值向量
min	矩阵各列的最小值向量
mean	矩阵各列的平均值向量
median	矩阵各列的中值向量
std	矩阵各列的标准偏差向量
sum	矩阵各列的元素之和向量
prod	矩阵各列的元素之积向量
cumsum	矩阵各列的累加和向量
cumprod	矩阵各列的累积积向量
diff	矩阵各列的差分向量
hist	矩阵各列的直方图向量
conv	列向量卷积
sort	排序
corrcoef	矩阵各列相关系数

对于向量变量，这些函数会把行向量和列向量都作为数组来处理。而对于矩阵变量，这些函数则面向矩阵的列进行运算，其结果是行向量。例如，函数 max 应用于矩阵变量时，返回的行向量的各元素的值恰为矩阵各列的最大值。用户可以自己编写面向列运算的函数 M 文件，只需加入一些辅助的条件语句判别出输入变量是向量还是矩阵。可以参考 MATLAB 的 M 文件 mean. m 和 diff. m。

注意，在 MATLAB 中，以上这些函数（除最后一个函数外）都可以应用于多维数组。例如，函数 sum 的一般调用命令为

》sum(M,d)

其中，M 是一个 n 维数组，d 可以是任何表达式，其值介于 1 与 n 之间。它的作用是在 n 维数组 M 的第 d 维方向计算 M 的元素之和，其结果应该是一个 $n-1$ 维数组。例如：

```
》M=zeros(3,3,4);
》for k=1: 4
M(:,:,k)=k * ones (3, 3);
end
》sum(M,1)
ans=
```

```
3     6     9     12
3     6     9     12
3     6     9     12
》sum(M,2)
ans=
 3     3     3
 6     6     6
 9     9     9
12    12    12
》sum(M, 3)
ans=
10    10    10
10    10    10
10    10    10
```

MATLAB 通过其矩阵运算，提供了强有力的数学运算功能，这正是 MATLAB 在化学计量学研究中最有用之处，一些常用而又十分重要的矩阵运算算法被直接设计成 MATLAB 的核心程序，而且运算速度很快，从而使得化学计量学的算法研究用 MATLAB 来编程变得十分简单。

矩阵分解是化学计量学研究的核心，很多算法大都基于矩阵分解，只要熟练掌握了矩阵分解的基本思路和基本算法，将为化学计量学的进一步研究打下了良好的基础。

1. 三角分解

矩阵的三角分解是最基本的一种矩阵分解方法，它将一个矩阵分解成上三角矩阵与下三角矩阵的乘积，称为 LU 分解或 LR 分解。实现 LU 分解的算法大都采用 Gauss 消去法，在 MATLAB 中，实现 LU 分解的函数是 lu，它的返回值是分解后的矩阵因子，调用命令为

```
》[L,U]=lu(A)
```

其中，U 是上三角矩阵，而 L 可以重新排列成一个下三角矩阵，使其主对角元素全为 1。这些矩阵因于可以用来计算矩阵的逆和矩阵行列式的值。LU 分解也是求解线性方程组的基本算法，另外，矩阵的除法以 LU 分解算法为基础。例如，设有矩阵

```
》A=[4 5 2;7 2 1;8 1 9]
```

即

```
A=
    4  5  2
    7  2  1
    8  1  9
```

对其进行 LU 分解，直接调用 lu 函数即可

```
》[L,U]=lu(A)
```

得

```
L=
    0.5000  1.0000       0
    0.8750  0.2500  1.0000
    1.0000       0       0
```

U=

```
    8.0000    1.0000    9.0000
         0    4.5000   -2.5000
         0         0   -6.2500
```

为了检验分解的正确性，计算两个矩阵因子之积的语句如下：

》L*U

计算结果为

```
ans=
  4    5    2
  7    2    1
  8    1    9
```

而矩阵 A 的逆矩阵为

》x=inv(A)

在 MATLAB 中，函数 inv 的算法之一是基于 LU 分解，也即 inv(A)=inv(U)*inv(L)。矩阵 A 的行列式为

》d=det(A)

-225

也可以由 LU 分解的矩阵因子的行列式得到

》d=det(L)*det(U)

得

```
d=
   -225
```

设有列向量 b=[1 3 5]，为了求解方程 Ax=b，利用 MATLAB 的矩阵除法，得到

》x=A\b

得

```
x=
  0.4756
 -0.2444
  0.1600
```

这个解也可以由 LU 分解得到

》y=L\b; x=U\y

还有一个与 LU 分解相关的 MATLAB 函数，即 chol。函数 chol 用于获得对称正定矩阵的 Cholesky 分解。

2. QR 分解

QR 分解可将一个方阵或长方矩阵表示成一个正交矩阵和一个上三角矩阵的乘积。例如：设

》A=[4 5 3;1 6 9;7 8 2;9 4 12]

得

A=

```
4  5   3
1  6   9
7  8   2
9  4  12
```

对其进行 QR 分解就可直接使用 qr 命令

```
》[Q,R]=qr(A)
```

其结果如下：

```
Q=
   -0.3299   -0.2630   -0.1666   -0.8912
   -0.0825   -0.7640    0.6246    0.1393
   -0.5774   -0.3500   -0.6000    0.4291
   -0.7423    0.4740    0.4713    0.0468
R=
  -12.1244   -9.7325   -11.7944
         0   -6.8029    -2.6769
         0         0     9.5773
         0         0          0
```

其中矩阵 Q 是正交的，而矩阵 R 是上三角形的，主对角线下的元素为零。通过下式可以验证矩阵 Q 的正交性

```
》Q*Q'
ans=
    1.0000  0.0000  0.0000  0.0000
    0.0000  1.0000  0.0000  0.0000
    0.0000  0.0000  1.0000  0.0000
    0.0000  0.0000  0.0000  1.0000
```

QR 分解还能为 MATLAB 函数 null 和 orth 提供基本算法。对一个矩阵而言，这两个函数分别给出矩阵变换的零空间和值域空向的正交基底。

3. 奇异值分解

奇异值分解在化学计量学中拥有十分重要的地位，化学计量学的很多核心方法都是基于奇异值分解，如主成分分析、主成分回归、偏最小二乘及大部分多元校正和多元分辨的方法都是基于奇异值分解，它是矩阵分析的一种有效的工具。有兴趣的读者可以参考 Golub 和 Van Loan 的著作《矩阵计算》以获得更多信息。在 MATLAB 中，下而的语句

```
》[U, S, V]=svd(A)
```

计算矩阵奇异值分解中的三个因子矩阵，它们满足条件

```
A=U*S*V'
```

其中，矩阵 U 是列正交矩阵和 V 是行正交矩阵，而 S 是对角矩阵。矩阵 U 和 V 在化学计量学中还分别被称为得分矩阵和载荷矩阵，它们分别张成矩阵 A 的列正交空间和行正交空间。在函数 svd 的调用式中，如果只有一个输出变量，则该函数返回的仅是矩阵 A 的奇异值。

使用矩阵奇异值分解算法的 MATLAB 函数主要有：矩阵的广义逆 pinv(A)、矩阵的秩 rank(A)、矩阵的 Euclide 范数 norm(A,2)和矩阵的条件数 cond(A)。

4. 特征值分解

设 A 是一个 $n\times n$ 阶矩阵，满足矩阵方程 $Ax=\alpha x$ 的 α 称为矩阵 A 的特征值。特征值可以由下列命令求得：

》eig(A)

它的返回值是矩阵 A 的特征值。如果矩阵 A 是一个实对称矩阵，那么所有的特征值都是实数。而下列形式的调用则可以同时得到特征值和特征向量：

》[x,D]=eig(A)

其中，矩阵 D 是由特征值组成的对角矩阵，矩阵 x 的列由对应的特征向量组成，使得方程

A * x=x * D 成立。

例如：

[x,d]=eig(A′ * A)

得

```
x=
     0.7460   -0.4074   0.5268
    -0.6600   -0.5579   0.5031
    -0.0889    0.7230   0.6851
d=
   25.5509         0          0
         0   54.8013          0
         0         0   445.6478
```

如果 A 和 B 都是方阵，那么 eig(A,B)的结果是使得方程 $Ax=kBx$ 成立的广义特征值组成的向量。为了同时得到相应的特征向量，可以用下列的命令形式：

》[x,D]=eig(A,B)

其中，D 是由特征值构成的对角矩阵，而矩阵 x 的列是由对应的特征向量组成的，满足方程

A * x=B * x * D。

四、MATLAB 的绘图功能

MATLAB 的绘图功能很强，它在我们进行程序调试和撰写论文时特别有用，如绘制三维图，等高线图等，可使一些在一般高级语言中难以实现的绘图功能，在 MATLAB 语言中，一个指令就可完成。在此我们也给出必要介绍。

首先，我们在此构造一个 80×80 的矩阵，然后用 MATLAB 语言来展示一下它的几种最常见的绘图能力（图 12-1～图 12-5）。

键入：

```
》A=[4 5 3 8
1 6 9 2
7 8 2 8
9 4 12 6]
```

得

A=

```
4  5   3  8
1  6   9  2
7  8   2  8
9  4  12  6
```

再键人

```
》B=[A A';1.2*A sqrt(A)]
》B1=[B 3*B';2.1*B B^2]
```

再键人

```
》plot(B1(:,9))
```

得

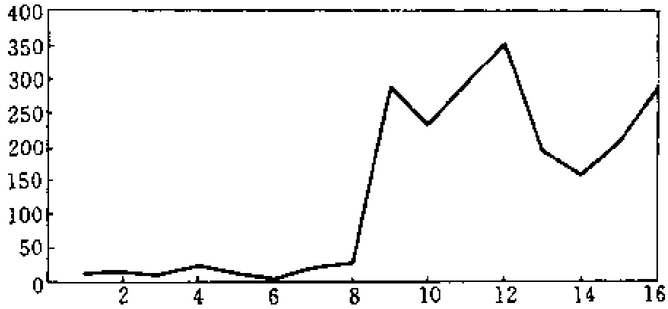


图 12-1

如继续键人

```
》subplot(221),plot(B1(:,9),'*-')
》hold
Current plot held
》subplot(222),plot(B1(:,10),'o-')
》subplot(223),plot(B1(:,11),'+-')
》subplot(224),plot(B1(:,11),'·-')
```

则可得

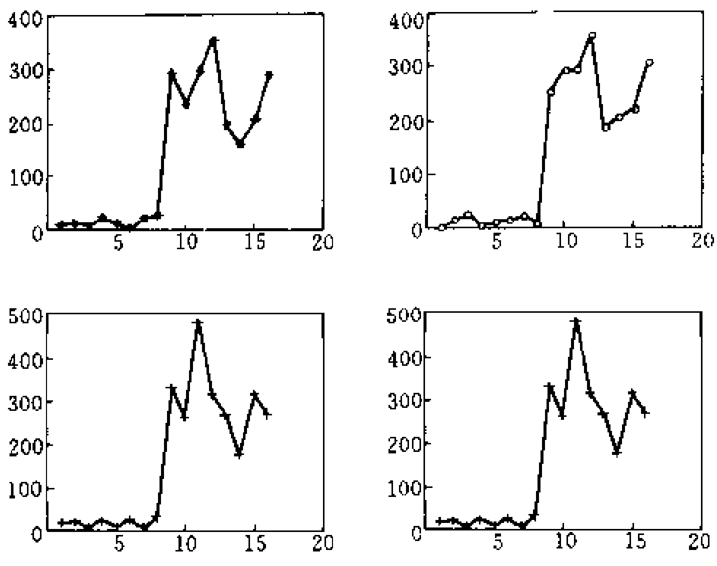


图 12-2

如键人

》bar(B1(:,6))

得

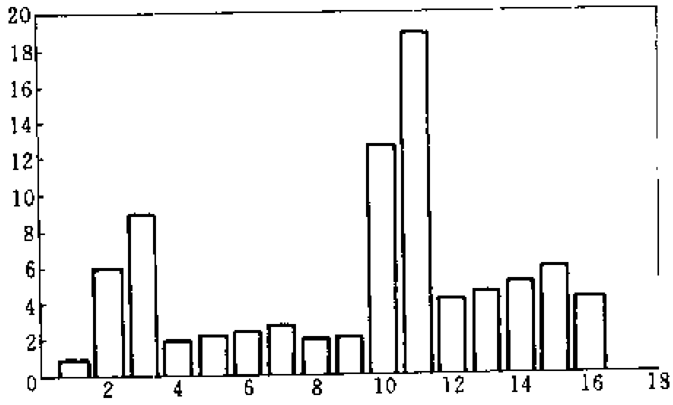


图 12-3

如键入

》contour(B1)

就立即可得等高线图

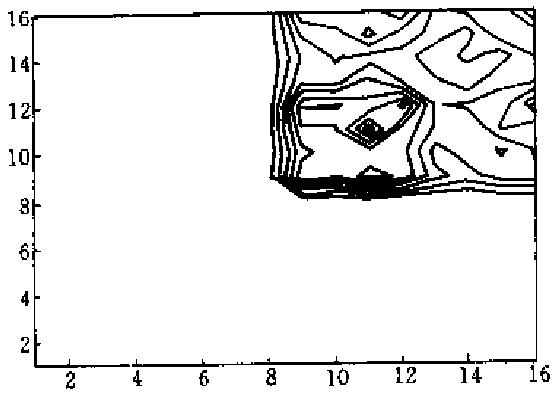


图 12-4

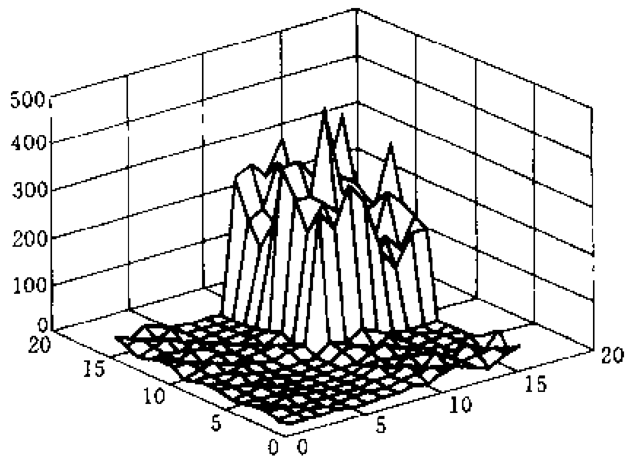


图 12-5

如键入

》mesh(B1)

就立即可得其三维图

第十三章 MATLAB 原程序目录及说明

1. function p = anova1(x,group)

函数 ANOVA1 可用于一维数据的方差分析（analysis of variance），主要用于比较 2 或 2 组数据以上的均值，其原假设（null hypothesis）是这些组数据的均值是相等的。

2. function [h,sig,ci] = ttest(x,m,alpha,tail)

函数 TTEST 主要用于将样本均值与一常数进行统计比较检验。

3. function [h, significance, ci] = ttest2(x,y,alpha,tail)

函数 TTEST2 主要用于将某样本均值与另一样本均值进行统计比较检验。

4. function x = ff2n(n)

函数 FF2N 用于构建两水平全因子设计表。X = FF2N（N）给出具有 N 列、2^n 行的设计矩阵 X，其中输入参数 N 为因子数。

5. function design = fullfact(levels)

函数 FULLFACT 用于混合水平（Mixed-level）的全因子设计矩阵。输入矢量参数 LEVELS 用于定义每列的水平数，如：LEVELS = [2 4 3]，函数 FULLFACT 将给出一个进行 24 次试验的全因子设计矩阵，其中第一列是 2 水平；第二列是 4 水平；第三列是 3 水平的设计。

6. function [settings, X] = cordexch(nfactors,nruns,model)

函数 CORDEXCH 用坐标交换（coordinate exchange）数值方法给出 D-最优设计表。

7. function [settings, X] = daugment(startdes,nruns,model)

函数 DAUGMENT 用坐标交换数值方法来增加原 D-最优试验设计的试验次数

8. function D = x2fx(x,model)

函数 X2FX 是一个用于函数 REGSTATS、函数 CORDEXCH 及函数 RSTOOL 的调用函数，它将执行将一个系统输入矩阵转换到一个给定回归模型的设计矩阵。

9. function [x, options] = fmins(funfcn,x,options,grad,varargin)

函数 FMINS 用于进行单纯形局部优化寻优。

10. function [x,OPTIONS] = fminu(FUN,x,OPTIONS,GRADFUN,varargin)

函数 fminu 通用局部最优算法函数，其预置算法采用 BFGS Quasi-Newton 法。

11. function [x,OPTIONS,lambda,HESS]=nlconst(FUNfcn,x,OPTIONS,VLB,VUB,GRADfcn,varargin);

函数 NLCONST 为下述函数的一个中间调用函数：

CONSTR，ATTGOAL，SEMINF 和 MINIMAX.

12. function [x,OPTIONS,lambda,HESS]=constr(FUN,x,OPTIONS,VLB,VUB,GRADFUN,varargin)

函数 CONSTR 用于对一个多变量函数的约束寻优。

13. function [obj_best,c_best]=vsgsa(DimN,region,belta0,delta_r,delta_r_limit);

函数 vsgsa 为变步长广义模拟退火算法子程序。

14. function obj=obj _ fun(location)

函数 obj _ fun 为变步长广义模拟退火算法中调用的一个自定义的具体目标函数。

15. function GLPoiX=glpoix(DimNum,PoiNum);

函数 GLPoiX 用于产生在（0，1）区间的好格子点。此程序可用于 2 或 3 为〉维空间的好格子点，并将结果在图上显示。

16. function [opti,reg]=sntopti(b,DimN,PoiN,y,x1,x2);

函数 sntopti 为序贯数论优化子程序。

17. function son=mate(popu)

函数 mate 为遗传算法中的交叉函数。

18. function son=mutate(popu,pmutate,mutatec)

函数 mutate 为遗传算法中的变异函数。

19. function parent=select(popufit)

函数 select 为遗传算法中的选择性交叉函数。

20. function offspr=compet(popufit,psize)

函数 compet 为遗传算法中的竞争函数。

21. function popufit=fitness(popuerr)

函数 popufit 为遗传算法中用于计算适应性的子程序；

22. function H = hadamard(n)

函数 HADAMARD 用于产生 Hadamard 矩阵。

23. function [b,bint,r,rint] = regress(y,X,alpha)

函数 REGRESS 用最小二乘方法来进行多元线性回归。

24. function regstats(responses,data,model)

函数 REGSTATS 用于对线性模型的回归诊断。该函数提供了一个对话界面，可得几乎所有单点诊断的信息。

25. function [cc,c,v]=kalmfil(x,y);

函数 kalmfil 可用于直接校正的 kalmam 估计

26. function [rig]=ridgecal(x,y);

函数 ridgecal 可用于岭回归的岭迹计算来估计病态分析体系的回归系数。

27. function [x,dx] = lscov(A,b,V)

函数 LSCOV 可用于直接校正的加权最小二乘估计。

28. function [p,cun]=pcr(x,c,numc,yun);

函数 pcr 为间接校正的主成分回归方法

29. function [p, q, w, b, t, u, x, y] = pls(a, c, maxrank)

函数 PLS 为偏最小二乘子程序

30. function [press,minlv] = Tcvpls(X,Y,maxnumb,selectnumb);

函数 Tcvpls 用于交叉效验（cross-validation）计算，本程序是调用的 PLS 子程序，同理，调用 PCR 也是可行的。

31. function [x,w] = nnls(E,f,tol)

函数 NNLS 用于解决非负最小二乘问题，即所得回归估计矢量为非负矢量。

32. function sida=m_ esti(x,y,iterNum);

函数 m _ esti 为稳健的 M _估计程序。

33. function [x,y]=simuld(numc,h1,cen1,sgm1,h2,cen2,sgm2,noise,c);

函数 simuld 可用于产生白色分析体系的模拟校正矩阵 x 和模拟混合物响应矢量 y。每一个组分光谱均由两个高斯峰所组成，而模拟混合物响应矢量 y 则由给定组分数（numc）的纯光谱由给定的浓度（c），通过线性加合而成。

34. function [eig]=rafa(x,y,alf _ range,stepwise);

函数 rafa 为一简易的广义秩消失因子分析法，其结果为一对数特征值对相对浓度系数的变化图，其对数特征值的极小点所对应的相对浓度系数值即为所求。

35. function [kons]=lorbm(x,s,c,rank);

函数 lorbm 为一广义秩消失因子分析法的 Lorber 方法，其结果为混合物中所含标准物质相对浓度系数值。

36. function [con]=rbl(x,y,ranky,num,noise)

函数 rbl 为残差双线性分解算法子程序。

37. function [bak1,bak2,bk1,bk2]=bakremo(action,x);

函数 bakremo 可用于 2 维数据的背景扣除，该程序应与函数 name 联合使用。该程序提供了一个图界面，用户就直接用鼠标在图上将峰簇前一段零组分区与峰簇后一段零组分区点出即可。

38. function [puprof,puspec]—als(x,pusini,iternum);

函数 als 为交换最小二乘（alterative least squares）子程序。该程序在迭代过程中提供了一个图界面，用户就直接用鼠标在图上将光谱或色谱中为负的部分点出即可。

39. function [eig,eig1]=evol(x,tim);

函数 evol 为渐进因子分析子程序。它将给出计算所得的前十个特征值的对保留时间对数变化图，以确定各组分的流出和消失点。

40. function [SS]=spc(X);

函数 spc 为子窗口因子分析方法子程序。其方法的基本思路和算法可参见第六章。

本程序提供了一个友好界面，用户只需用鼠标定义两个相邻多组分区域，且该两个多组分区 域都含有一个共同组分，则可得到该组分的纯光谱。

41. function [pure]=Orthopro(x);

函数 Ortbopro 为正交投影分辨子程序。其方法的基本思路和算法可参见第五章。

本程序提供了一个友好界面，用户只需用鼠标定义两个相邻多组分区域，且该两个多组分区域都含有一个共同组分，则可得到该组分的纯光谱。

42. function b=lsq(x,y)

函数 lsq 为简单最小二乘计算程序，主要用于其他函数调用。

43. function [x]=movpca(action,x,in1)

函数 movpca 可用于动态局部主成分分析，通过滑动条即可对数据矩阵的任一部分进行固定窗口的主成分分析。

44. function name

函数 name 为一与函数 bakremo，函数 movpca 共同使用的程序，以保证全局变量的安全使用。使用该函数时，将要用的 2 维数据矩阵名以字符串输入即可。

45. function [scores,loads,ssq,res] = pca(data,plots,scl,lv)

函数 pca 用于主成分分析计算。

46. function [C,S]=rescomp(X,indices);

函数 rescomp 为在每个化学组分都有选择性信息时的分辨程序。

47. function [C,S,Xred]=respart(X,indices,numc);

函数 respart 为在部分化学组分有选择性信息时的分辨程序。

48. function indices=scoop(X);

函数 scoop 可用于通过特征投影图来找到组分的选择性信息。这样得到的信息可用于下一步的分辨。

49. function y=selnorm(x,thres);

函数 selnorm 用于对数据矩阵进行选择性标准化，以消除不等性方差的对体系组分数估计的影响。

50. function indices =seqrank(X,noiselimit);

函数 seqrank 可用于特征结构跟踪分析，以得到保留时间方向的化学秩图。该程序提供了一个图界面，用户就直接用鼠标在图上将可能的选择性区域点出，程序就可对该区域进行计算，并通过对话，要用户确定它是否是选择性区域，一经确定，程序将自动将该区域计下。

51. function [simu,y,y1]=Simu2D

函数 Simu2D 为产生二维数据的计算机模拟程序。最高可产生包含多组分混合体系。本程序采用对话形式来输入模拟所需参数。每个光谱由两个高斯峰所组成，每个色谱由一个高斯峰所组成。

52. Zoom

函数 Zoom 为二维作图缩放功能函数。

53. function class = classify(sample,training,group)

函数 CLASSIFY 为线性判别分析子程序。

54. function d = mahal(Y,X);

函数 MAHAL 为计算 Mahalanobis 距离的子程序。

55. function [center, U, obj _ fcn] = fcm(data, cluster _ n, options)

函数 FCM 为采用模糊 K-均值聚类方法对样本矩阵进行聚类子程序。

56. function U = initfcm(cluster _ n, data _ n)

函数 INITFCM 产生模糊 K-均值聚类方法的初始分配矩阵子程序。

57. function [U _ new, center, obj _ fcn] = stepfcm(data, U, cluster _ n, expo)

函数 STEPFCM 为模糊 K-均值聚类方法中迭代一步计算子程序。

58. function out = distfcm(center, data)

函数 DISTFCM 可用于模糊 K-均值聚类方法中计算数据矩阵中样本点到类中心的欧氏(Euclidean) 距离。

59. function [timeseri,y]=Rung _ kuta(x0,time _ range,stepwise);

函数 Rung _ kuta 为四阶 Runge-Kutta 法；

60. function [timeseri,y]=Adambm(x0,time _ range,stepwise);

函数 Adambm 为求解为常微分组的初值问题的 Adams 预估_校正方法；

61. function [timeseri,y]=Hamming(x0,time _ range,stepwise);

函数 Hamming 为求解为常微分组的初值问题的 Hamming 定步长方法；

62. function f=dery(y)

函数 dery 为定义一个常微分方程组的调用函数。

63. function [tout,yout,o3,o4,o5,o6] = ode45(odefile,tspan,y0,options,varargin)

函数 ODE45 为解决非刚性常微分方程的初值问题的通用程序。

64. function [tout,yout,o3,o4,o5,o6] = ode15s(odefile,tspan,y0,options,varargin)

函数 ODE15S 为解决刚性常微分方程的初值问题的通用程序。

65. function a = appcoef(c,l,arg3,arg4,arg5)

函数 APPCOEF 用于计算一维信号的逼近系数。

66. function F = coifwavf(wname)

函数 COIFWAVF 用于计算 Coiflets 小波滤波器.

67. function w = dbaux(N,sumw);

函数 DBAUX 用于计算 Daubechies 小波滤波器

68. function F = dbwavf(wname)

函数 DBWAVF 用于计算 Daubechies 小波滤波器.

69. function [thr,sorh,keepapp,crit] = ddencmp(dorc,worwp,x)

函数 DDENCMP 用于计算去噪或压缩预置值。

70. function d = detcoef(c,l,n)

函数 DETCOEF 用于计算一维信号的细节系数 (detail coefficients).

71. function [a,d] = dwt(x,arg2,arg3)

函数 DWT 用于单水平一维小波变换。

72. function [a,d] = dwtper(x,arg2,arg3)

函数 DWTPER 用于计算单水平离散一维小波变换 (周期性的) (periodized).

73. function y = dyaddown(x,arg2,arg3)

函数 DYADDOWN 计算二分的下采样 (Dyadic downsampling)

74. function y = dyadup(x,arg2,arg3)

函数 DYADUP 计算二分的上采样 (Dyadic upsampling), 它通过简单填零而使该过程在小波重构算法中非常有用。

75. function x = idwt(a,d,arg3,arg4,arg5)

函数 IDWT 计算单水平逆离散一维小波变换。它根据不同小波基函数 ('wname', 参见 WFILTERS), 执行单水平逆离散一维小波重构 (reconstruction)

76. function x = idwtper(a,d,arg3,arg4,arg5)

用于计算单水平离散一维小波变换(周期性的)(periodized).

函数 IDWTPER 用于计算单水平逆离散一维小波变换 (周期性的) (periodized).

77. function thr = thrwdaid(c,l,tptr,scal,n)

函数 THRWDAID 用小波变换计算一维自动去噪的阈值。

78. function thr = thselect(x,tptr)

函数 THSELECT 可用于选择去噪的阈值.

79. function y = upcoef(o,x,arg3,arg4,arg5,arg6)

函数 UPCOEF 可用于用一维小波系数来直接重构分析信号。

80. function [c,l,a] = upwlev(c,l,arg3,arg4)

函数 UPWLEV 用于一维信号小波分解的单水平重构。

81. function [c,l] = wavedec(x,n,arg3,arg4)

函数 WAVEDEC 可用于计算多水平一维小波分解。它通过或是小波基函数或是小波分解滤波器来计算多水平一维小波分解

82. function thr = wdcbm(c,l,alpha,m)

函数 WDCBM 用 Birge-Massart 策略对一维分析信号去噪和压缩

83. function [xd,cxd,lxd] = wden(in1,in2,in3,in4,in5,in6,in7)

函数 WDEN 用小波来自动对一维分析信号进行去噪。

84. function [xc,cxc,lxc,perf0,perfl2] =…

wdencmp(o,in2,in3,in4,in5,in6,in7,in8)

函数 WDENCMP 可用于用小波进行去噪和压缩。它可对一分析信号或图像来进行去噪和压缩。

85. function [out1,out2,out3,out4,out5,out6,out7,out8] = wfilters(wname,o)

函数 WFILTERS 计算小波滤波器（Wavelet filters）.

86. function y = wkeep(x,arg2,arg3)

函数 WKEEP 可用于保存一个矢量或一个矩阵的一部分。

87. function [x,xn] = wnoise(num,n,snrat,init)

函数 WNOISE 可用于产生一个含噪声的小波去噪的检测矢量。

88. function stdc = wnoisest(c,l,s)

函数 WNOISEST 可用于估计一维小波系数矢量的噪声。

89. function x = wrcoef(o,c,l,arg4,arg5,arg6)

函数 WRCOEF 用于从一维小波分解结构（C 和 L），或特定小波基函数（'wname'），或特定重构滤波器（Lo _ R and Hi _ R）进行重构系数矢量计算。

90. function m = wrmcoef(o,c,l,arg4,arg5,arg6)

函数 WRMCOEF 用于从一维小波分解结构得到重构系数矢量矩阵。

91. function y = wthresh(x,sorh,t)

函数 WTHRESH 执行软（soft）或硬（hard）取阈值.

92. function [a1,a2,a3] = simuff(p,w1,b1,f1,w2,h2,f2,w3,b3,f3)

函数 SIMUFF 用于模拟前传网络。SIMUFF 函数最高可模拟三层网络。

93. function [w1,b1,w2,b2,w3,b3] = initff(p,s1,f1,s2,f2,s3,f3)

函数 INITFF Inititialize 对三层及以下的前传网络进行初始化。

94. function [a,b,c,d,e,f,g,h] = trainbp(i,j,k,l,m,n,o,p,q,r,s,t)

函数 TRAINBP 用误差反传来训练前传网络。TRAINBP 函数可调用权重和偏置来训练三层前传网络。

95. function [a,b,c,d,e,f,g,h] = trainbpx(i,j,k,l,m,n,o,p,q,r,s,t)

函数 TRAINBPX 采用快速误差反传来训练前传网络，可以训练一层、二层，最高可以训练到三层网络。

96. function [a,b,c,d,e,f,g,h] = trainbpm(i,j,k,l,m,n,o,p,q,r,s,t)

函数 TRAINBPM 采用误差反传加惯量常数来训练前传网络，可以训练一层、二层，最高可以训练到三层网络。

97. function [a,b,c,d,e,f,g,h] = trainlm(i,j,k,l,m,n,o,p,q,r,s,t)

函数 TRAINLM 采用 Levenberg-Marquardt 方法来训练前传网络，可以训练一层、二层，最高可以训练到三层网络。

98. function n = purelin(n,b)

函数 PURELIN 为线性转换或称活性函数.

99. function a = tansig(n,b)

函数 TANSIG 为双曲线正弦 S 形转换或称活性函数。

100. function a = logsig(n,b)

函数 LOGSIG 为对数 S 形（sigmoid）转换函数。

101. function d = deltalin(a,d,w)

函数 DELTALIN PURELIN 神经元的 Delta 函数。

102. function d = deltatan(a,d,w)

函数 DELTATAN TANSIG 神经元的 Delta 函数.

103. function d = deltalog(a,d,w)

函数 DELTALOG LOGSIG 神经元的 Delta 函数。

104. function [dw,db] = learnbp(p,d,lr)

函数 LEARNBP 为误差反传学习规则函数。

105. function [dw,db] — learnbm(p,d,lr,mc,dw,db)

函数 LEARNBPM 为含有惯量因子的误差反传学习规则函数。

106. function j = learnlm(p,d)

函数 LEARNLM 为 Levenberg-Marquardt 学习规则函数。

107. function m = errsurf(p,t,wv,bv,f)

函数 ERRSURF 为误差表面函数.

108. function plotes(wv,bv,es,v)

函数 PLOTES 为误差表面作图函数。

109. function hh = plotep(w,b,e,h)

函数 PLOTEP 为权重和偏置在误差表面位置作图函数。

110. function [w1,b1,i,tr] = tbp1(w1,b1,f1,p,t,tp,wv,bv,es,v)

函数 TBP1 用于训练一层前传误差反传网络。

111. function [w1,b1,w2,b2,i,tr]=tbp2(w1,b1,f1,w2,b2,f2,p,t,tp)

函数 TBP2 用于训练二层前传误差反传网络。

112. function [w1,b1,w2,b2,w3,b3,i,tr] = tbp3(w1,b1,f1,w2,b2,f2,w3,b3,f3,p,t,tp)

函数 TBP3 用于训练三层前传误差反传网络

113. function [w1,b1,i,tr] = tbpx1(w1,b1,f1,p,t,tp)

函数 TBPX1 用于训练一层网络的快速误差反传方法。

114. function [w1,b1,w2,b2,i,tr] = tbpx2(w1,b1,f1,w2,b2,f2,p,t,tp)

函数 TBPX2 用于训练二层网络的快速误差反传方法。

115. function [w1,b1,w2,b2,w3,b3,i,tr]=tbpx3(w1,b1,f1,w2,b2,f2,w3,b3,f3,p,t,tp)

函数 TBPX3 用于训练三层网络的快速误差反传方法。

116. function [w1,b1,i,tr] = tlm1(w1,b1,f1,p,t,tp)

函数 TLM1 训练一层前传误差反传网络的 Levenberg-Marquardt 方法.

117. function [w1,b1,w2,b2,i,tr] = tlm2(w1,b1,f1,w2,b2,f2,p,t,tp)

函数 TLM2 采用 Levenberg-Marquardt 方法，用于训练二层前传误差反传网络。

118. function [w1,b1,w2,b2,w3,b3,i,tr] = tlm3(w1,b1,f1,w2,b2,f2,w3,b3,f3,p,t,tp)

函数 TLM3 采用 Levenberg-Marquardt 方法，用于训练三层前向误差反传网络。

119. function lmsobj=lmsobj(c,x,y);

函数 lmsobj 为最小方差中位数目标函数子程序，为函数 sntopti 调用子程序。

第十四章　MATLAB 原程序

MATLAB 语言在进行矩阵运算方面，显得特别简捷、高效和方便。在化学计量学的研究和应用中，由于大都是进行多变量运算，矢量和矩阵是基本运算单元，所以，MATLAB 语言很快成为其研究的主要工具，目前国际上两本化学计量学专业杂志，"Journal of Chemometris"和"Chemometrics and Intelligent Laboratory Systems"多次对其进行了专门的介绍，使其几乎成为进行化学计量学研究的特定计算机编程语言。随着 MATLAB 版本的不断更新，其功能越来越强，使之在诸如一般数值计算、化学计量学、数字信号处理、系统识别、自动控制、振动理论、时序分析与建模、优化设计、人工神经网络、动态仿真系统、小波分析、特殊函数和图形等领域，表现出一般计算机高级语言难以比拟的优势。所以，本书所给出的计算机程序一般都是 MATLAB 语言的程序，其中特别是有关人工神经网络和小波变换部分，大都是从 MATLAB 软件包(5.1 版本)的工具库中直接得到，在给出一些基本程序的基础上，还给出了一些怎样在计算机上进行练习和使用的说明。其余有关一些别的程序，大部分是本实验室编出，并在计算机上多次使用通过。对于使用这些程序的方法及输入与输出变量也给出了必要的说明。对于所列计算机程序，我们也尽量给出其调用时所需的函数子程序，但对于一些 MATLAB 语言的最基本函数子程序，建议读者参阅相关的专门介绍 MATLAB 语言的书籍。

本书给出的 MATLAB 语言计算机程序，一般是按章节编排先后给出的。当然，有些程序可在几个不同章节中使用。对于这一点，在本附录就没特别说明了。

1. function p = anova1(x,group)

函数 ANOVA1 可用于一维数据的方差分析 (analysis of variance)，主要用于比较 2 或 2 组数据以上的均值，其原假设 (null hypothesis) 是这些组数据的均值是相等的。

【例子】

(1) P = ANOVA1(X)

其中输入参数 X 为一数据矩阵，其每一列视为一组数据。此程序适用于平衡方差分析，即每组数据量相等的情况。

输出参数 P 为检验原假设计算所得的值。

(2) P = ANOVA1(X,GROUP)

其中输入参数 X 在此必须为一矢量，而另一矢量参数 GROUP 则表示在矢量 X 中数据的元素相对应属于哪一组，这样，每组的数据量就没有数目必须相等的限制了。

输出参数 P 意义同前。

程序 ANOVA1 还同时给出两个图：一个图给出标准的一维方差分析表；另一个图给出形象表示各组数据均值及方差大小的箱图 (boxplot)。

参考文献：

Robert V. Hogg, and Johannes Ledolter, Engineering Statistics, Macmillan 1987 pp. 205～206

【程序】

```
if nargin == 2
  lx = length(x);
  if lx ~= prod(size(x))
```

```
  error('Requires a vector first argument.')
end

li = length(group);
if li ~= lx
  error('The inputs must be the same length.');
end

if any(group) < 1 | any(floor(group)) ~= group
  error('The second argument must be a positive integer.');
end
x = x(:);
group = group(:);
maxi = max(group);
xm = zeros(1,maxi);
countx = xm;
for j = 1:maxi
  k = find(group == j);
  if j == 1
    M = x(k);
  else
    [r, c] = size(M);
    lk = length(k);
    if lk > r
  tmp = NaN;
      M(r+1:lk,:) = tmp(ones(lk-r,c));
      tmp = x(k);
      M = [M tmp];
    else
      tmp = x(k);
      tmp1 = NaN;
      tmp((lk + 1):r,1) = tmp1(ones(r-lk,1));
      M = [M tmp];
    end
  end
    countx(j) = length(k);
    xm(j) = mean(x(k));             % column means
  end

  gm = mean(x);               % grand mean
  df1 = maxi- 1;              % Column degrees of freedom
  df2 = lx-maxi;              % Error degrees of freedom
  RSS = countx .* (xm-gm) * (xm-gm)'; % Regression Sum of Squares
```

```
end

if nargin == 1
   [r,c] = size(x);
   xm = mean(x);                 % column means
   gm = mean(xm);                % grand mean
   df1 = c-1;                % Column degrees of freedom
   df2 = c * (r-1);              % Error degrees of freedom
   RSS = r * (xm - gm) * (xm-gm)';              % Regression Sum of Squares
end

TSS = (x(:) - gm)' * (x(:) - gm);              % Total Sum of Squares
SSE = TSS - RSS;               % Error Sum of Squares

if (SSE~=0)
   F = (RSS/df1) / (SSE/df2);
   p = 1 - fcdf(F,df1,df2);              % Probability of F given equal means.
elseif (RSS==0)              % Constant Matrix case.
   F = 0;
   p = 1;
else              % Perfect fit case.
   F = Inf;
   p = 0;
end

Table=zeros(3,4);                %Formatting for ANOVA Table printout
Table(:,1)=[ RSS SSE TSS]';
Table(:,2)=[df1 df2 df1+df2]';
Table(:,3)=[ RSS/df1 SSE/df2 Inf ]';
Table(:,4)=[ F Inf Inf ]';

fig1 = figure('pos', [50 350 500 300]);
set(gca,'Visible','off');

z0 = 0.00;
y0 = 0.85;
dz = 0.15;
dy =0.06;
text(z0+0.40,y0,'ANOVA Table')
z=z0+dz;
y=y0-3 * dy/2;
colheads = ['Source     ';'        SS ';'        df ';'        MS ';'        F  '];

for i=1:5
```

```
    text(z,y,colheads(i,:))
    z = z + dz;
  end
  rowheads = ['Columns    ';'Error      ';'Total      '];
  for i=1:3
    y = y0-(i-1.5)*dy;
    z = z0 - dz;
    h = text(z,y,rowheads(i,:));

    z = z - dz;
    for j=1:4
    if (Table(i,j) ~= Inf) & j ~=2,
      h = text(z,y,sprintf('%11.4g',Table(i,j)));
      z = z + dz;
    elseif j==2,
      z = z + dz/4;
      h = text(z,y,sprintf('%7.5g',Table(i,j)));
      z = z + 3*dz/4;
    end
  end
end

fig2 = figure('pos',get(gcf,'pos') + [0,-200,0,0]);

if nargin == 1
  boxplot(x,1);
else
  boxplot(M,1);
  h = get(gca,'XLabel');
  set(h,'String','Group Number');
end
```

2. function [h,sig,ci] = ttest(x,m,alpha,tail)

函数 TTEST 主要用于将样本均值与一常数进行统计比较检验。

【例子】

[H,SIG] = TTEST(X,M,ALPHA,TAIL)

对一个服从正态分布的随机变量执行均值检验。

输入参数:

X 服从正态分布的随机变量量测值;

M 为假设常数;

ALPHA 为显著性水平 (significance level) 参数;

TAIL 为检验选择参数，当 TAIL=0 时，表示备选假设为“均值等于 M”(双边检验)；当 TAIL=1 时，表示备选假设为“均值大于 M”(单边检验)；

当 TAIL=-1 时，表示备选假设为“均值小于 M”(单边检验)；

M = 0, ALPHA = 0.05 和 TAIL = 0 为预置值。

输出参数：

SIG 为原假设对被检验数据而言其为真的概率；

H 为结果参数，当 H=0 时，表示在给定的显著性水平上不拒绝原假设；而当 H=1 时，表示在给定的显著性水平上拒绝原假设. 1 of alpha."

参考文献：

E. Kreyszig, "Introductory Mathematical Statistics", John Wiley, 1970, page 206

【程序】

```
if nargin < 1,
  error('Requires at least one input argument.');
end

[m1 n1] = size(x);
if (m1 ~= 1 & n1 ~= 1)
  error('Requires a vector first input argument.');
end

if nargin < 2
  m = 0;
end
if nargin < 4,
  tail = 0;
end
if nargin < 3,
  alpha = 0.05;
end

if (alpha <= 0 | alpha >= 1)
  fprintf('Warning: significance level must be between 0 and 1\n');
  h = NaN;
  sig = NaN;
  ci = [NaN NaN];
  return;
end

samplesize = length(x);
xmean = mean(x);
ser = std(x) ./ sqrt(samplesize);
tval = (xmean - m) / ser;
sig = tcdf(tval,samplesize - 1);

% the significance just found is for the tail = -1 test
crit = tinv(1 - alpha,samplesize - 1) .* ser;
if tail == 1
  sig = 1 - sig;
```

```
elseif tail == 0
  sig = 2 * min(sig,1 - sig);
  crit = tinv((1 - alpha / 2),samplesize - 1) .* ser;
end
ci = [(xmean - crit) (xmean + crit)];

% Determine if the actual significance exceeds the desired significance
h = 0;
if sig <= alpha,
  h = 1;
end

if isnan(sig),
  h = NaN;
end
```

3. function [h, significance, ci] = ttest2(x,y,alpha,tail)

函数 TTEST2 主要用于将某样本均值与另一样本均值进行统计比较检验。

【例子】

[H,SIGNIFICANCE CI] = TTEST2(X,Y,ALPHA,TAIL)

对两个服从正态分布的随机变量执行均值检验。

输入参数：

X 服从正态分布的随机变量量测值；

M 为假设常数；

ALPHA 为显著性水平（significance level）参数；

TAIL 为检验选择参数，当 TAIL=0 时，表示备选假设为"随机变量 X 的均值等于随机变量 Y"（双边检验）；当 TAIL=1 时，表示备选假设为"随机变量 X 的均值大于随机变量 Y"（单边检验）；当 TAIL=1 时，表示备选假设为"随机变量 X 的均值小于随机变量 Y"（单边检验）；

M = 0. ALPHA = 0.05 和 TAIL = 0 为预置值。

输出参数：

SIGNIFICANCE 为原假设对被检验数据而言其为真的概率；

H 为结果参数，当 H=0 时，表示在给定的显著性水平上不拒绝原假设；而当 H=1 时，表示在给定的显著性水平上拒绝原假设.

参考文献：

E. Kreyszig, "Introductory Mathematical Statistics", John Wiley, 1970, section 13.4. (Table 13.4.1 on page 210)

【程序】

```
if nargin < 2,
  error('Requires at least two input arguments');
end

[m1 n1] = size(x);
[m2 n2] = size(y);
if (m1 ~= 1 & n1 ~= 1) | (m2 ~= 1 & n2 ~= 1)
  error('Requires vector first and second inputs.');
end
```

```
if nargin < 4,
   tail = 0;
end

if nargin < 3,
   alpha = 0.05;
end

if (alpha <= 0 | alpha >= 1)
   fprintf('Warning: significance level must be between 0 and 1\n');
   h = NaN;
   sig = NaN;
   return;
end

dfx = length(x) - 1;
dfy = length(y) - 1;
dfe = dfx + dfy;
msx = dfx * var(x);
msy = dfy * var(y);

difference = mean(x) - mean(y);
pooleds = sqrt((msx + msy) * (1/(dfx + 1) + 1/(dfy + 1)) / dfe);

ratio = difference / pooleds;

criticalvalue = tinv(1 - alpha / 2,dfe);
spread = criticalvalue * pooleds;

ci = [(difference - spread) (difference + spread)];

% Find the significance probability for the tail = 1 test.
significance = 1 - tcdf(ratio,dfe);
% Adjust the significance probability for other null hypotheses.
if tail == -1
   significance = 1 - significance;
elseif tail == 0
   significance = 2 * min(significance,1 - significance);
end

% Determine if the actual significance exceeds the desired significance
h = 0;
if significance <= alpha,
```

```
  h = 1;
end

if isnan(significance),
  h = NaN;
end
```

4. function x = ff2n(n)

函数 FF2N 用于构建两水平全因子设计表。X = FF2N(N)给出具有 N 列、2^N行的设计矩阵 X,其中输入参数 N 为因子数。

【程序】

```
rows = 2.^(n);
ncycles = rows;
group = zeros(rows,n);

for k = 1:n
  settings = (0:1);
  ncycles = ncycles/2;
  nreps = rows./(2*ncycles);
  settings = settings(ones(1,nreps),:);
  settings = settings(:);
  settings = settings(:,ones(1,ncycles));
  x(:,n-k+1) = settings(:);
end
```

5. function design = fullfact(levels)

函数 FULLFACT 用于混合水平 (Mixed-level) 的全因子设计矩阵。输入矢量参数 LEVELS 用于定义每列的水平数，如：LEVELS = [2 4 3]，函数 FULLFACT 将给出一个进行 24 次试验的全因子设计矩阵，其中第一列是 2 水平；第二列是 4 水平；第三列是 3 水平的设计。

【程序】

```
[m,n] = size(levels);

if min(m,n) ~= 1
  error('Requires a vector input.');
end

if any((floor(levels) - levels) ~= 0) | any(levels <= 1)
  error('The input values must be integers greater than one.');
end

ssize = prod(levels);
ncycles = ssize;
cols = max(m,n);

design = zeros(ssize,cols);
```

```
for k = 1:cols
  settings=(1:levels(k));
  ncycles = ncycles./levels(k);
  nreps = ssize./(ncycles * levels(k));
  settings = settings(ones(1,nreps),:);
  settings = settings(:);
  settings = settings(:,ones(1,ncycles));
  design(:,k) = settings(:);
end
```

6. function [settings, X] = cordexch(nfactors,nruns,model)

函数 CORDEXCH 用坐标交换(coordinate exchange)数值方法给出 D-最优设计表。

【例子】

[SETTINGS, X] = CORDEXCH(NFACTORS,NRUNS,MODEL)

输入参数：

NFACTORS 表示试验的因子数；

NRUNS 表示想要进行试验的次数；

字符串参数 MODEL 表示所选择的回归模型，其预置值为一个包括常数项的线性加合模型；

当 MODEL='interaction'时，表示一个包括常数项，线性项和交叉项的回归模型；

当 MODEL='quadratic'时，表示一个包括常数项、线性项、交叉项以及平方项的回归模型；

当 MODEL='purequadratic'时，表示一个包括常数项、线性项、和平方项的回归模型；

输出参数：

SETTINGS 为一因子设置矩阵；

X 为相应的设计矩阵。

【程序】

```
  settings = unifrnd(-1,1,nruns,nfactors);
if nargin == 2
  model = 'linear';
end

if ~isstr(model)
  if nfactors ~= size(model,2)
    error('The number of columns in a numeric model matrix must equal the number of factors.');
  end

  nxij = linspace(-1,1,max(model(:))-1)';
else
  nxij = (-1:1)';
end

X = x2fx(settings,model);
if size(X,2) > size(X,1)
  error('There are not enough runs to fit the specified model.');
end
```

```
[Q,R]=qr(X,0);
logdetX = 2 * sum(log(abs(diag(R))));

maxiter = 10;
iter = 0;
madeswitch = 1;

% Create Iteration Counter Figure.
screen = get(0,'ScreenSize');
f = figure('Units','Pixels','Position',[25 screen(4)-150 300 60],...
    'Color',[0.5 0.5 0.5]);
ax = gca;
set(ax,'Visible','off');
t=text(0,0.7,'Coordinate exchange iteration counter');
set(t,'FontName','Geneva','FontSize',12);

while madeswitch > 0 & iter < maxiter
  madeswitch = 0;
  iter = iter + 1;

  % Update iteration counter.
  s=sprintf('Iteration %4i',iter);
  set(f,'CurrentAxes',ax);
  h=text(0,0.2,s);
  set(h,'FontName','Geneva','FontSize',12);
  drawnow;

  %Loop over rows of factor settings matrix.
  for row = 1:nruns
    %Loop over columns of factor settings matrix.
    for col = 1:nfactors
      fx = x2fx(settings(row,:),model);
      xij = settings(row,col);
      xnew = settings(row,:);
      xnew = xnew(ones(length(nxij),1),:);
      xnew(:,col) = nxij;
      fxnew = x2fx(xnew,model);
      % Compute change in determinant.
      fxold = fx(ones(length(nxij),1),:);
      E = fxold/R;
      F = fxnew/R;
      dxold = sum((E.*E)')';
      dxnew = sum((F.*F)')';
      dxno = sum((E.*F)')';
```

```
        d = (1 - dxnew). * (1 - dxold) + dxno. ^ 2;

        % Find the maximum change in the determinant.
        [d,idx] = max(d);
        % Switch coordinates if the maximum change is greater than 1.
        if d > 1,
          madeswitch = 1;
          logdetX = log(d) + logdetX;
          settings(row,:) = xnew(idx,:);
          X(row,:) = fxnew(idx,:);
          [Q,R] = qr(X,0);
        end
      end
    end
delete(h);
end
close(f);
```

7. function [settings, X] = daugment(startdes,nruns,model)

函数 DAUGMENT 用坐标交换数值方法来增加原 D-最优试验设计的试验次数

【例子】

[SETTINGS, X] = DAUGMENT(STARTDES,NRUNS,MODEL)

输入参数:

NFACTORS 表示试验的因子数;

NRUNS 表示想要进行试验的次数;

字符串参数 MODEL 表示所选择的回归模型，其预置值为一个包括常数项的线性加合模型;

当 MODEL='interaction'时，表示一个包括常数项，线性项和交叉项的回归模型;

当 MODEL='quadratic'时，表示一个包括常数项、线性项、交叉项以及平方项的回归模型;

当 MODEL='purequadratic'时，表示一个包括常数项、线性项、和平方项的回归模型;

输出参数:

SETTINGS 为一因子设置矩阵;

X 为相应的设计矩阵。

【程序】

```
[nobs, nfactors] = size (startdes);

settings = unifrnd (-1, 1, nruns, nfactors);
if nargin == 2
  model = 'linear';
end

settings = [startdes; settings];

X = x2fx (settings, model);
[Q, R] =qr (X, 0);
```

```
logdetX = 2 * sum (log (abs (diag (R))));

info = inv (X' * X);

% Create Iteration Counter Figure.
screen = get (0,'ScreenSize');
f = figure ('Units','Pixels','Position', [25 screen (4) -150 300 60], ...
    'Color', [0.5 0.5 0.5]);
ax = gca;
set (ax,'Visible','off');
t=text (0, 0.7,'D-optimal design iteration counter');
set (t,'FontName','Geneva','FontSize', 12);

maxiter = 10;
iter = 0;
madeswitch = 1;

while madeswitch > 0 & iter < maxiter
  madeswitch = 0;
  iter = iter + 1;

  % Update iteration counter.
  s=sprintf ('Iteration %4i', iter);
  set (f,'CurrentAxes', ax);
  h=text (0, 0.2, s);
  set (h,'FontName','Geneva','FontSize', 12);
  drawnow;

  for row = nobs+1: nobs+nruns
    for col = 1: nfactors
      fx = x2fx (settings (row,:), model);
        xij = settings (row, col);
      nxij = (-1: 1)';
      xnew = settings (row,:);
      xnew = xnew (ones (length (nxij), 1),:);
      xnew (:, col) = nxij;
      fxnew = x2fx (xnew, model);
      fxold = fx (ones (length (nxij), 1),:);

      % Compute change in determinant.
      E = fxold/R;
        F = fxnew/R;
        dxold = sum ( (E. * E)')';
        dxnew = sum ( (F. * F)')';
```

```
        dxno = sum ( (E. *F)')';

        d = (1 + dxnew). * (1 - dxold) - dxno. ^2;

        % Find the maximum change in the determinant.
       [d, idx] = max (d);
      if d > 1,
         madeswitch = 1;
         logdetX = log (d) + logdetX;
        settings (row,:) = xnew (idx,:);
        X (row,:) = fxnew (idx,:);
         [Q, R] = qr (X, 0);
      end
    end
  end
  delete (h);
end
close (f);
```

8. function D = x2fx(x,model)

函数 X2FX 是一个用于函数 REGSTATS、函数 CORDEXCH 及函数 RSTOOL 的调用函数，它将执行将一个系统输入矩阵转换到一个给定回归模型的设计矩阵。

【例子】

D = X2FX(X,'MODEL')

输入参数：

X 为系统输入矩阵；

字符串参数 MODEL 表示所选择的回归模型，其预置值为一个包括常数项的线性加合模型；

当 MODEL='interaction'时，表示一个包括常数项，线性项和交叉项的回归模型；

当 MODEL='quadratic'时，表示一个包括常数项、线性项、交叉项以及平方项的回归模型；

当 MODEL='purequadratic'时，表示一个包括常数项、线性项、和平方项的回归模型；

输出参数：

D 为一个给定回归模型的设计矩阵。

参见 RSTOOL, CORDEXCH, ROWEXCH, REGSTATS.

【程序】

```
[m, n] = size (x);
if nargin == 1,
  D = [ones (m, 1) x];
  return,
end
if strcmp (model,'linear') | strcmp (model,'l') | strcmp (model,'L') | ...
  strcmp (model,'Linear') | strcmp (model,'additive') | strcmp (model,'Additive'),
  D = [ones (m, 1) x];

elseif strcmp (model,'interaction') | strcmp (model,'quadratic') | ...
  strcmp (model,'q') | strcmp (model,'i') | strcmp (model,'Interaction') | ...
```

```
  strcmp (model,'Quadratic') | strcmp (model,'interactions') | ...
  strcmp (model,'Interactions') | strcmp (model,'Q') | strcmp (model,'I'),
  D = [ones (m, 1) x];
  for idx = 1: n
    xidx = x (:, idx);
    xidx = xidx (:, ones (n-idx, 1));
    D = [D xidx .  * x (:, idx+1: n)];
  end
  if strcmp (model,'quadratic') | strcmp (model,'q') | strcmp (model,'Q') | ...
    strcmp (model,'Quadratic'),
    D = [D x.  * x];
  end

elseif strcmp (model,'purequadratic') | strcmp (model,'p') | ...
   strcmp (model,'Purequadratic') | strcmp (model,'P'),
   D = [ones (m, 1) x x.  * x];

elseif strcmp (model,'model') | strcmp (model,'MODEL')
   dlgname = 'Not a Model';
   warningstr = ['Sorry, ' model ' is not a valid string for the model parameter.'];
   warndlg (warningstr, dlgname);
elseif isstr (model)
   D = feval (model, x); %Allows for extensions to named higher order models. (e. g. 'cubic')
elseif ~isstr (model)
   [row, col] = size (model);
   if col ~= n
     error ('A numeric second argument must have the same number of columns as the first argument.');
   end
   D = zeros (m, row);
   for idx = 1: row
     carat = model (idx,:);
     tmp = x.  ^ carat (ones (m, 1),:);
     if size (tmp, 2) == 1
       D (:, idx) = tmp;
     else
       D (:, idx) = (prod (tmp'))';
     end
   end
end
```

9. function [x, options] = fmins(funfcn,x,options,grad,varargin)

函数 FMINS 用于进行单纯形局部优化寻优。

【例子】

(1) X = FMINS('F',X0)

输入参数:

F 为一字符串参数,用以表示目标函数;

如:F = 'exp(x(1)) * (4 * x(1)^2 + 2 * x(2)^2 - 4 * x(1) * x(2) + 2 * x(2) + 1)';

或为一以 M 文件表示的函数,参见使用示例;

X0 为初始搜索坐标位置;

输出参数:

X 为最终坐标位置;

(2) [X,OPTIONS] = FMINS('F',X0,OPTIONS,[],P1,P2,…)

输入参数:

F 为一字符串参数,用以表示目标函数;如:

F = 'exp(x(1)) * (4 * x(1)^2 + 2 * x(2)^2 + 4 * x(1) * x(2) + 2 * x(2) + 1)';

或为一以 M 文件表示的函数,参见使用示例;

X0 为初始搜索坐标位置;

OPTIONS 为最优化计算工具箱通用选择参数,参见有关它的说明

P1,P2,…:为以 M 文件表示的目标函数所需调用的参数;

输出参数:

X 为最终坐标位置;

OPTIONS 为最优化计算工具箱通用选择参数,参见有关它的说明

说明:函数 FMINS 采用 Nelder-Mead 的单纯形搜索优化法。

参见 FMIN, FOPTIONS.

参考文献:

J. E. Dennis, Jr. and D. J. Woods, New Computing Environments: Microcomputers in Large-Scale Computing, edited by A. Wouk, SIAM, 1987, pp. 116~122

【程序】

```
if nargin<3, options = []; end
options = foptions(options);
prnt = options(1);
tol = options(2);
tol2 = options(3);
% The input argument grad is there for compatability with FMINU in
% the Optimization Toolbox, but is not used by this function.

% Convert to inline function as needed.
funfcn = fcnchk(funfcn,length(varargin));
n = prod(size(x));
if (~options(14))
  options(14) = 200 * n;
end

% Set up a simplex near the initial guess.
xin = x(:); % Force xin to be a column vector
v = xin; % Place input guess in the simplex! (credit L. Pfeffer at Stanford)
x(:) = v; fv = feval(funfcn,x,varargin{:});
```

```
% Following improvement suggested by L. Pfeffer at Stanford
usual _ delta = 0.05;                % 5 percent deltas for non-zero terms
zero _ term _ delta = 0.00025;         % Even smaller delta for zero elements of x
for j = 1:n
  y = xin;
  if y(j) ~= 0
    y(j) = (1 + usual _ delta) * y(j);
  else
    y(j) = zero _ term _ delta;
  end
  v = [v y];
  x(:) = y; f = feval(funfcn,x,varargin{:});
  fv = [fv f];
end
[fv,j] = sort(fv);
v = v(:,j);

func _ evals = n+1;
if prnt > 0
  clc
  format compact
  format short e
  home
  func _ evals
  disp('initial ')
  disp('         ')
  v
  f
end

alpha = 1; beta = 1/2; gamma = 2;
[n,np1] = size(v);
onesn = ones(1,n);
ot = 2:n+1;
on = 1:n;

% Iterate until the diameter of the simplex is less than tol.
while func _ evals < options(14)
    if max(max(abs(v(:,ot)-v(:,onesn)))) <= tol & ...
          max(abs(fv(1)-fv(ot))) <= tol2
        break
    end

    % One step of the Nelder-Mead simplex algorithm
```

```
  vbar = (sum(v(:,on)')/n)';
  vr = (1 + alpha) * vbar - alpha * v(:,n+1);
  x(:) = vr;
  fr = feval(funfcn,x,varargin{:});
  func_evals = func_evals + 1;
  vk = vr; fk = fr; how = 'reflect ';
  if fr < fv(n)
    if fr < fv(1)
    ve = gamma * vr + (1-gamma) * vbar;
    x(:) = ve;
    fe = feval(funfcn,x,varargin{:});
    func_evals = func_evals + 1;
    if fe < fv(1)
      vk = ve; fk = fe;
      how = 'expand ';
    end
  end
else
  vt = v(:,n+1); ft = fv(n+1);
  if fr < ft
    vt = vr; ft = fr;
  end
  vc = beta * vt + (1-beta) * vbar;
  x(:) = vc;
  fc = feval(funfcn,x,varargin{:});
  func_evals = func_evals + 1;
  if fc < fv(n)
    vk = vc; fk = fc;
    how = 'contract';
  else
    for j = 2:n
      v(:,j) = (v(:,1) + v(:,j))/2;
      x(:) = v(:,j);
      fv(j) = feval(funfcn,x,varargin{:});
    end
  func_evals = func_evals + n-1;
  vk = (v(:,1) + v(:,n+1))/2;
  x(:) = vk;
  fk = feval(funfcn,x,varargin{:});
  func_evals = func_evals + 1;
  how = 'shrink ';
  end
end
```

```
  v(:,n+1) = vk;
  fv(n+1) = fk;
  [fv,j] = sort(fv);
  v = v(:,j);

  if prnt > 0
    func_evals
    disp(how)
    disp('        ')
    v
    fv
  end
end
x(:) = v(:,1);
if prnt > 0, format, end
options(10)=func_evals;
options(8)=min(fv);
if func_evals==options(14)
  if options(1) >= 0
    disp(['Warning: Maximum number of function evaluations (',...
        int2str(options(14)),') has been exceeded']);
    disp('        (increase OPTIONS(14)).')
  end
end
```

10. function [x,OPTIONS] = fminu(FUN,x,OPTIONS,GRADFUN,varargin)

函数 fminu 通用局部最优算法函数,其预置算法采用 BFGS Quasi-Newton 法。

【例子】

(1) X=FMINU('FUN',X0)

输入参数:

FUN 为一字符串参数,用以表示目标函数,如:

FUN = 'exp(x(1)) * (4 * x(1)^2 + 2 * x(2)^2 + 4 * x(1) * x(2) + 2 * x(2) + 1)';

或为一以 M 文件表示的函数,参见使用示例;

X0 为初始搜索坐标位置;

输出参数:

X 为最终坐标位置;

(2) [X,OPTIONS]=FMINU('FUN',X0,OPTIONS,'GRADFUN',P1,P2,...)

输入参数:

FUN 为一字符串参数,用以表示目标函数;如:

FUN= 'exp(x(1)) * (4 * x(1)^2 + 2 * x(2)^2 + 4 * x(1) * x(2) + 2 * x(2) + 1)';

或为一以 M 文件表示的函数,参见使用示例;

X0 为初始搜索坐标位置;

OPTIONS 为最优化计算工具箱通用选择参数,参见有关它的说明;

GRADFUN 为调用一个名为'GRADFUN'的函数以得到某给定函数 f 的在点 X 之上的导数 df/dX.

P1,P2,...:为以 M 文件表示的目标函数所需调用的参数。

输出参数：

X 为最终坐标位置；

OPTIONS 为最优化计算工具箱通用选择参数，参见有关它的说明。

说明：函数 FMINU 预置采用 BFGS Quasi-Newton 法，以一个混合 2 阶和 3 阶多项式进行一维搜索过程(line search procedure)。

【程序】

```
% --------------------------------Initialization --------------- ---------------
XOUT=x(:);
nvars=length(XOUT);

if nargin < 2, error('fminu requires two input arguments');end
if nargin < 3, OPTIONS=[]; end
if nargin < 4, GRADFUN=[]; end

% Convert to inline function as needed.
if ~isempty(FUN)
  [funfcn, msg] = fcnchk(FUN,length(varargin));
  if ~isempty(msg)
    error(msg);
  end
else
  error('FUN must be a function name or valid expression.')
end

if ~isempty(GRADFUN)
  [gradfcn, msg] = fcnchk(GRADFUN,length(varargin));
  if ~isempty(msg)
    error(msg);
  end
else
  gradfcn = [];
end

f = feval(funfcn,x,varargin{:});
n = length(XOUT);
GRAD=zeros(nvars,1);
OLDX=XOUT;
MATX=zeros(3,1);
MATL=[f;0;0];
OLDF=f;
FIRSTF=f;
[OLDX,OLDF,HESS,OPTIONS]=optint(XOUT,f,OPTIONS);
CHG = 1e-7*abs(XOUT)+1e-7*ones(nvars,1);
SD = zeros(nvars,1);
```

```
diff = zeros(nvars,1);
PCNT = 0;
how = '      ';

OPTIONS(10)=2; % Function evaluation count (add 1 for last evaluation)
status =-1;

while status ~= 1
% Work Out Gradients
    if isempty(gradfcn) | OPTIONS(9)
       OLDF=f;
% Finite difference perturbation levels
% First check perturbation level is not less than search direction.
       f = find(10 * abs(CHG)>abs(SD));
       CHG(f) = -0.1 * SD(f);
% Ensure within user-defined limits
       CHG = sign(CHG+eps). * min(max(abs(CHG),OPTIONS(16)),OPTIONS(17));
       for gcnt=1:nvars
          XOUT(gcnt,1)=XOUT(gcnt)+CHG(gcnt);
          x(:) = XOUT;
          f = feval(funfcn,x,varargin{:});
          GRAD(gcnt)=(f -OLDF)/(CHG(gcnt));
          if f < OLDF
             OLDF=f;
          else
             XOUT(gcnt)=XOUT(gcnt)-CHG(gcnt);
          end
       end
% Try to set difference to 1e-8 for next iteration
% Add eps for machines that can't handle divide by zero.
       CHG = 1e-8./(GRAD + eps);
       f = OLDF;
       OPTIONS(10)=OPTIONS(10)+nvars;
% Gradient check
       if OPTIONS(9) == 1
          GRADFD = GRAD;
          x(:)=XOUT;
          GRAD(:) = feval(gradfcn,x,varargin{:});
          if isa(gradfcn,'inline')
             graderr(GRADFD, GRAD, formula(gradfcn));
          else
             graderr(GRADFD, GRAD, gradfcn);
          end
           OPTIONS(9) = 0;
```

```
        end

      else
        OPTIONS(11)=OPTIONS(11)+1;
        x(:)=XOUT;
        GRAD(:) = feval(gradfcn,x,varargin{:});
      end
%--------------------Initialization of Search Direction---------------- -----
if status == -1
    SD=-GRAD;
    FIRSTF=f;
    OLDG=GRAD;
    GDOLD=GRAD' * SD;
% For initial step-size guess assume the minimum is at zero.
    OPTIONS(18) = max(0.001, min([1,2 * abs(f/GDOLD)]));
    if OPTIONS(1)>0
       disp([sprintf('%5.0f  %12.6g  %12.6g ',OPTIONS(10),f,OPTIONS(18)),sprintf('%12.3g ',
GDOLD)]);
    end
    XOUT=XOUT+OPTIONS(18) * SD;
    status=4;
    if OPTIONS(7)==0; PCNT=1; end

else
%------------Direction Update------------------
    gdnew=GRAD' * SD;
    if OPTIONS(1)>0,
       num=[sprintf('%5.0f  %12.6g  %12.6g ',OPTIONS(10),f,OPTIONS(18)),sprintf('%12.3g ',
gdnew)];
    end
    if (gdnew>0 & f>FIRSTF)|~finite(f)
% Case 1: New function is bigger than last and gradient w.r.t. SD -ve
%     ...interpolate.
       how='inter';
       [stepsize]=cubici1(f,FIRSTF,gdnew,GDOLD,OPTIONS(18));
       if stepsize<0|isnan(stepsize), stepsize=OPTIONS(18)/2; how='C1f '; end
       if OPTIONS(18)<0.1&OPTIONS(6)==0
          if stepsize * norm(SD)<eps
             stepsize=exp(rand(1,1)-1)-0.1;
             how='RANDOM STEPLENGTH';
             status=0;
          else
             stepsize=stepsize/2;
          end
```

```
        end
        OPTIONS(18)=stepsize;
        XOUT=OLDX;
    elseif f<FIRSTF
        [newstep,fbest] =cubici3(f,FIRSTF,gdnew,GDOLD,OPTIONS(18));
        sk=(XOUT-OLDX)' * (GRAD-OLDG);
        if sk>1e-20
% Case 2: New function less than old fun. and OK for updating HESS
%         .... update and calculate new direction.
        how='        ';
          if gdnew<0
            how='incstep';
            if newstep<OPTIONS(18), newstep=2 * OPTIONS(18)+1e-5;
            how=[how,' IF'];
            end
            OPTIONS(18)=min([max([2,1.5 * OPTIONS(18)]),1+sk+abs(gdnew)+max([0,OPTIONS
(18)-1]), (1.2+0.3 * (~OPTIONS(7))) * abs(newstep)]);
          else % gdnew>0
            if OPTIONS(18)>0.9
              how='int_st';
              OPTIONS(18)=min([1,abs(newstep)]);
            end
          end %if gdnew
          [HESS,SD]=updhess(XOUT,OLDX,GRAD,OLDG,HESS,OPTIONS);
          gdnew=GRAD' * SD;
          OLDX=XOUT;
          status=4;
% Save Variables for next update
          FIRSTF=f;
          OLDG=GRAD;
          GDOLD=gdnew;
% If mixed interpolation set PCNT
          if OPTIONS(7)==0, PCNT=1; MATX=zeros(3,1); MATL(1)=f; end
      elseif gdnew>0 %sk<=0
% Case 3: No good for updating HESSIAN .. interpolate or halve step length.
          how='inter_st';
          if OPTIONS(18)>0.01
            OPTIONS(18)=0.9 * newstep;
            XOUT=OLDX;
          end
          if OPTIONS(18)>1, OPTIONS(18)=1; end
        else
% Increase step, replace starting point
          OPTIONS(18)=max([min([newstep-OPTIONS(18),3]),0.5 * OPTIONS(18)]);
```

```
        how='incst2';
        OLDX=XOUT;
        FIRSTF=f;
        OLDG=GRAD;
        GDOLD=GRAD' * SD;
          OLDX=XOUT;
      end % if sk>
% Case 4: New function bigger than old but gradient in on
%         ...reduce step length.
    else %gdnew<0 & F>FIRSTF
      if gdnew<0&f>FIRSTF
        how='red _ step';
        if norm(GRAD-OLDG)<1e-10; HESS=eye(nvars); end
        if abs(OPTIONS(18))<eps
          SD=norm(nvars,1) * (rand(nvars,1)-0.5);
          OPTIONS(18)=abs(rand(1,1)-0.5) * 1e-6;
                how='RANDOM SD';
              else
                OPTIONS(18)=-OPTIONS(18)/2;
        end
        XOUT=OLDX;
      end %gdnew>0
    end % if (gdnew>0 & F>FIRSTF)|~finite(F)
    XOUT=XOUT+OPTIONS(18) * SD;
    if isinf(OPTIONS(1))
      disp([num,how])
    elseif OPTIONS(1)>0
      disp(num)
    end
end %----------End of Direction Update-------------------

% Check Termination
    if max(abs(SD))<2 * OPTIONS(2) & (-GRAD' * SD) < 2 * OPTIONS(3)
      if OPTIONS(1) > 0
        disp('Optimization Terminated Successfully')
        disp(' Search direction less than 2 * options(2)')
        disp(' Gradient in the search direction less than 2 * options(3)')
        disp([' NUMBER OF FUNCTION EVALUATIONS=',
int2str(OPTIONS(10))]);
      end
      status=1;
    elseif OPTIONS(10)>OPTIONS(14)
        if OPTIONS(1) >= 0
            disp('Maximum number of function evaluations exceeded;')
```

```
            disp(' increase options(14).')
        end
        status=1;
    else

% Line search using mixed polynomial interpolation and extrapolation.
      if PCNT~=0
        while PCNT > 0 & OPTIONS(10) <= OPTIONS(14)
          x(:) = XOUT;
          f = feval(funfcn,x,varargin{:});
          OPTIONS(10)=OPTIONS(10)+1;
          [PCNT,MATL,MATX,steplen,f,
how]=searchq(PCNT,f,OLDX,MATL,MATX,SD,GDOLD,OPTIONS(18), how);
          OPTIONS(18)=steplen;
          XOUT=OLDX+steplen * SD;
        end
        if OPTIONS(10)>OPTIONS(14)
        if OPTIONS(1) >= 0
          disp('Maximum number of function evaluations exceeded;')
          disp(' increase options(14).')
        end
        status=1;
      end
    else
      x(:)=XOUT;
      f = feval(funfcn,x,varargin{:});
      OPTIONS(10)=OPTIONS(10)+1;
    end
  end
end

x(:)=XOUT;
f = feval(funfcn,x,varargin{:});
if f > FIRSTF
  OPTIONS(8) = FIRSTF;
  x(:)=OLDX;
else
  OPTIONS(8) = f;
end
```

11. function

```
[x,OPTIONS,lambda,HESS]=nlconst(FUNfcn,x,OPTIONS,VLB,VUB,GRADfcn,vara rgin);
  %
  函数 NLCONST 为下述函数的一个中间调用函数:
                CONSTR, ATTGOAL, SEMINF 和 MINIMAX.
```

【程序】

```
global OPT _ STOP OPT _ STEP;
OPT _ STEP = 1;
OPT _ STOP = 0;
% Initialize so if OPT _ STOP these have values
lambda = []; HESS = [];

Nlconst = 'nlconst';

% Set up parameters.
XOUT=x(:);

VLB=VLB(:); lenvlb=length(VLB);
VUB=VUB(:); lenvub=length(VUB);
bestf = Inf;

nvars = length(XOUT);

CHG = 1e-7 * abs(XOUT)+1e-7 * ones(nvars,1);
if lenvlb * lenvlb>0
    if any(VLB( (1:lenvub)' ) > VUB), error('Bounds Infeasible'), end
end
for i=1:lenvlb
    if lenvlb>0,if XOUT(i)<VLB(i),XOUT(i)=VLB(i)+1e-4; end,end
end
for i=1:lenvub
    if lenvub>0,if XOUT(i)>VUB(i),XOUT(i)=VUB(i);CHG(i)=-CHG(i);end,end
end

% Used for semi-infinite optimization:
s = nan; POINT =[]; NEWLAMBDA =[]; LAMBDA = []; NPOINT =[]; FLAG = 2;
OLDLAMBDA = [];
x(:) = XOUT; % Set x to have user expected size
% Compute the objective function and constraints
if strcmp(FUNfcn{4},'seminf')
  [f,g,NPOINT,NEWLAMBDA,OLDLAMBDA,LOLD,s] = ...
    semifun(x,LAMBDA,NEWLAMBDA,OLDLAMBDA,POINT,FLAG,s,varargin{:});
else
  [f,g,msg] = opteval(x,FUNfcn,varargin{:});
  error(msg);
  g = g(:);
end
if isempty(f)
  error('FUN must return a non-empty objective function.')
```

```
end
ncstr = length(g);

% Evaluate gradients and check size
if isempty(GRADfcn)
  analytic_gradient = 0;
else
  analytic_gradient = 1;
  [gf_user,gg_user,msg] = opteval(x,GRADfcn,varargin{:});
  error(msg);
  gf_user = gf_user(:);
  % Both might evaluate to empty when expression syntax is used
  if isempty(gf_user) & isempty(gg_user)
    analytic_gradient = 0;
  else % Either gf or gg is defined
    if length(gf_user) ~= nvars
      error('The objective gradient is the wrong size.')
    end
    if isempty(gg_user) & isempty(g)
      % Make gg compatible
      gg = g'; gg_user = g';
    else % Check size of gg
      [ggrow, ggcol] = size(gg_user);
      if ggrow ~= nvars
        error('The constraint gradient has the wrong number of rows.')
      end
      if ggcol ~= ncstr
        error('The constraint gradient has the wrong number of columns.')
      end
    end % isempty(gg_user)
  end % isempty(gf_user) & isempty(gg_user)
end % isempty(GRADfcn)

OLDX=XOUT;
OLDG=g;
OLDgf=zeros(nvars,1);
gf=zeros(nvars,1);
OLDAN=zeros(ncstr,nvars);
LAMBDA=zeros(ncstr,1);
sizep = length(OPTIONS);
OPTIONS = foptions(OPTIONS);
if lenvlb * lenvlb>0
      if any(VLB((1:lenvub)') > VUB), error('Bounds Infeasible'), end
end
```

```
for i=1:lenvlb
        if lenvlb>0,if XOUT(i)<VLB(i),XOUT(i)=VLB(i)+eps; end,end
end
OPTIONS(18)=1;
if OPTIONS(1)>0
   if OPTIONS(7)==1
        disp('         ')
        disp('f--COUNT     MAX{g}        STEP   Procedures');
   else
   disp('          ')
        disp('f--COUNT     FUNCTION        MAX{g}        STEP   Procedures');
   end
end
HESS=eye(nvars,nvars);
if sizep<1 |OPTIONS(14)==0, OPTIONS(14)=nvars * 100;end
OPTIONS(10)=1;
OPTIONS(11)=1;
GNEW=1e8 * CHG;

%------------------Main Loop------------------
status = 0;
while status ~= 1

%------------------GRADIENTS------------------

   if ~analytic_gradient | OPTIONS(9)
% Finite Difference gradients (even if just checking analytical)
      POINT = NPOINT;
      oldf = f;
      oldg = g;
      ncstr = length(g);
      FLAG = 0; % For semi-infinite
      gg = zeros(nvars, ncstr); % For semi-infinite
% Try to make the finite differences equal to 1e-8.
      CHG = -1e-8./(GNEW+eps);
      CHG = sign(CHG+eps).* min(max(abs(CHG),OPTIONS(16)),OPTIONS(17));
      OPT_STEP = 1;
      for gcnt=1:nvars
        if gcnt == nvars,
          FLAG = -1;
        end
        temp = XOUT(gcnt);
        XOUT(gcnt)= temp + CHG(gcnt);
        x(:) =XOUT;
```

```
        if strcmp(FUNfcn{4},'seminf')
          [f,g,NPOINT,NEWLAMBDA,OLDLAMBDA,LOLD,s] = ...
            semifun(x,LAMBDA,NEWLAMBDA,OLDLAMBDA,POINT,FLAG,s,varargin{:});
        else
          [f,g,msg] = opteval(x,FUNfcn,varargin{:});
          error(msg);
          g = g(:);
        end
        OPT_STEP = 0;
        if OPT_STOP
          break;
        end
        % Next line used for problems with varying number of constraints
        if ncstr~=length(g),
          diff=length(g);
          g=v2sort(oldg,g);
        end

        gf(gcnt,1) = (f-oldf)/CHG(gcnt);
        if ~isempty(g)
          gg(gcnt,:) = (g - oldg)'/CHG(gcnt);
        end
        XOUT(gcnt) = temp;
      end % for
      if OPT_STOP
        break;
      end

% Gradient check
      if OPTIONS(9) == 1 & analytic_gradient
        gfFD = gf;
        ggFD = gg;
        gg = gg_user;
        gf = gf_user;

        disp('Function derivative')
        if isa(GRADfcn{1},'inline')
          graderr(gfFD, gf, formula(GRADfcn{1}));
        else
          graderr(gfFD, gf, GRADfcn{1});
        end
        if ~isempty(gg)
          disp('Constraint derivative')
```

```
        if isa(GRADfcn{3},'inline')
          graderr(ggFD, gg, formula(GRADfcn{3}));
        else
          graderr(ggFD, gg, GRADfcn{3});
        end
      end
      OPTIONS(9)=0;
    end % OPTIONS(9) == 1 & analytic_gradient
    FLAG=1: % For semi-infinite
    OPTIONS(10)=OPTIONS(10) + nvars;
    f=oldf;
    g=oldg;
  else % analytic_gradient & options(9)=0
    % User-supplied gradients
    % gf and gg already computed first time through loop
    if OPTIONS(10) > 1
      gg=zeros(nvars, ncstr);
      [gf,gg,msg]=opteval(x,GRADfcn,varargin{:});
      error(msg);
      gf=gf(:);
      if isempty(gg) & isempty(g)
        gg=g';
      end
    else
      % First time through loop
      gg=gg_user;
      gf=gf_user;
    end

    if OPT_STOP
      break;
    end

  end   % if ~analytic_gradient | OPTIONS(9)
  AN=gg';
  how='        ';
  OPT_STEP=2;

%-------------SEARCH DIRECTION---------------

  for i=1:OPTIONS(13)
    schg=AN(i,:) * gf;
    if schg>0
      AN(i,:)=-AN(i,:);
```

```
      g(i)= -g(i);
    end
  end

  if OPTIONS(11)>1   % Check for first call
    % For equality constraints make gradient face in
    % opposite direction to function gradient.
    if OPTIONS(7)~=5,
      NEWLAMBDA=LAMBDA;
    end
    [ma,na]=size(AN);
    GNEW=gf + AN' * NEWLAMBDA;
    GOLD=OLDgf+OLDAN' * LAMBDA;
    YL=GNEW-GOLD;
    sdiff=XOUT-OLDX;
    % Make sure Hessian is positive definite in update.
    if YL' * sdiff<OPTIONS(18)^ 2 * 1e-3
      while YL' * sdiff<-1e-5
         [YMAX,YIND]=min(YL. * sdiff);
         YL(YIND)=YL(YIND)/2;
      end
      if YL' * sdiff < (eps * norm(HESS,'fro'));
         how=' Hessian modified twice';
         FACTOR=AN' * g - OLDAN' * OLDG;
         FACTOR=FACTOR. * (sdiff. * FACTOR>0). * (YL. * sdiff<=eps);
         WT=1e-2;
         if max(abs(FACTOR))==0; FACTOR=1e-5 * sign(sdiff); end
           while YL' * sdiff < (eps * norm(HESS,'fro')) & WT < 1/eps
               YL=YL+WT * FACTOR;
               WT=WT * 2;
           end
         else
           how=' Hessian modified';
      end
    end

%----------Perform BFGS Update If YL'S Is Positive---------
    if YL' * sdiff>eps
      HESS=HESS ...
     +(YL * YL')/(YL' * sdiff)-
((HESS * sdiff) * (sdiff' * HESS'))/(sdiff' * HESS * sdiff);
     %  BFGS Update using Cholesky factorization of Gill, Murray and Wright.
     %  In practice this was less robust than above method and slower.
     %  R=chol(HESS);
```

```
    %   s2=R * S; y=R'\YL;
    %   W=eye(nvars,nvars)-(s2' * s2)\(s2 * s2') - (y' * s2)\(y * y');
    %   HESS=R' * W * R;
   else
    how=' Hessian not updated';
   end

else % First call
   OLDLAMBDA=(eps+gf' * gf) * ones(ncstr,1)./(sum(AN'.* AN')'+eps);
end % if OPTIONS(11)>1
OPTIONS(11)=OPTIONS(11)+1;

LOLD=LAMBDA;
OLDAN=AN;
OLDgf=gf;
OLDG=g;
OLDF=f;
OLDX=XOUT;
XN=zeros(nvars,1);
if (OPTIONS(7)>0&OPTIONS(7)<5)
   % Minimax and attgoal problems have special Hessian:
   HESS(nvars,1:nvars)=zeros(1,nvars);
   HESS(1:nvars,nvars)=zeros(nvars,1);
   HESS(nvars,nvars)=1e-8 * norm(HESS,'inf');
   XN(nvars)=max(g); % Make a feasible solution for qp
 end
 if lenvlb>0,
    AN=[AN; -eye(lenvlb,nvars)];
    GT=[g;-XOUT((1:lenvlb)')+VLB];
 else
    GT=g;
 end
 if lenvub>0
    AN=[AN;eye(lenvub,nvars)];
    GT=[GT;XOUT((1:lenvub)')-VUB];
 end
 HESS=(HESS + HESS') * 0.5;
[SD,lambda,howqp]=qpsub(HESS,gf,AN,-GT,[],[],XN,OPTIONS(13),-1, ...
                                Nlconst,size(AN,1),nvars);

lambda((1:OPTIONS(13))')=abs(lambda( (1:OPTIONS(13))' ));
ga=[abs(g( (1:OPTIONS(13))' )) ; g( (OPTIONS(13)+1:ncstr)' ) ];
if ~isempty(g)
   mg=max(ga);
```

```
else
   mg=0;
end

if OPTIONS(1)>0
   if strncmp(howqp,'ok',2);
      howqp ='        ';
   end
      if ~isempty(how) & ~isempty(howqp)
         how=[how,'; '];
      end
      if OPTIONS(7)==1,
         gamma=mg+f;
         disp([sprintf('%5.0f %12.6g ',OPTIONS(10),gamma), ...
              sprintf('%12.3g  ',OPTIONS(18)),how, '       ',howqp]);
      else
         disp([sprintf('%5.0f %12.6g %12.6g ',OPTIONS(10),f,mg), ...
              sprintf('%12.3g  ',OPTIONS(18)),how, '       ',howqp]);
      end
end
LAMBDA=lambda((1:ncstr)');
OLDLAMBDA=max([LAMBDA';0.5 * (LAMBDA+OLDLAMBDA)'])' ;

%---------------LINESEARCH--------------------
MATX=XOUT;
MATL=f+sum(OLDLAMBDA.*(ga>0).*ga) + 1e-30;
infeas=strncmp(howqp,'i',1);
if OPTIONS(7)==0 | OPTIONS(7) == 5
     % This merit function looks for improvement in either the constraint
     % or the objective function unless the sub-problem is infeasible in which
     % case only a reduction in the maximum constraint is tolerated.
     % This less "stringent" merit function has produced faster convergence in
     % a large number of problems.
     if mg > 0
      MATL2=mg;
     elseif f >=0
      MATL2=-1/(f+1);
     else
      MATL2=0;
     end
     if ~infeas & f < 0
      MATL2=MATL2 + f - 1;
     end
  else
```

```
  % Merit function used for MINIMAX or ATTGOAL problems.
  MATL2=mg+f;
end
if mg < eps & f < bestf
  bestf=f;
  bestx=XOUT;
  bestHess=HESS;
  bestlambda=lambda;
end
MERIT=MATL + 1;
MERIT2=MATL2 + 1;
OPTIONS(18)=2;
while(MERIT2 > MATL2) & (MERIT > MATL) ...
     & OPTIONS(10) < OPTIONS(14) & ~OPT_STOP
  OPTIONS(18)=OPTIONS(18)/2;
  if OPTIONS(18) < 1e-4,
    OPTIONS(18)= -OPTIONS(18);

    % Semi-infinite may have changing sampling interval
    % so avoid too stringent check for improvement
    if OPTIONS(7) == 5,
      OPTIONS(18)=-OPTIONS(18);
      MATL2=MATL2 + 10;
    end
  end
  XOUT=MATX + OPTIONS(18) * SD;
  x(:)=XOUT;

  if strcmp(FUNfcn{4},'seminf')
      [f,g,NPOINT,NEWLAMBDA,OLDLAMBDA,LOLD,s]=...
          semifun(x,LAMBDA,NEWLAMBDA,OLDLAMBDA,POINT,FLAG,s,varargin{:});
  else
    [f,g,msg]=opteval(x,FUNfcn,varargin{:});
    error(msg);
    g=g(:);
  end
  if OPT_STOP
    break;
  end

  OPTIONS(10)=OPTIONS(10) + 1;
  ga=[abs(g( (1:OPTIONS(13))' )) ; g( (OPTIONS(13)+1:length(g))' )];
  if ~isempty(g)
    mg=max(ga);
```

```
    else
      mg=0;
    end

    MERIT=f+sum(OLDLAMBDA. * (ga>0). * ga);
    if OPTIONS(7)==0 | OPTIONS(7) == 5
      if mg > 0
        MERIT2=mg;
      elseif f >=0
        MERIT2=-1/(f+1);
      else
        MERIT2=0;
      end
      if ~infeas & f < 0
        MERIT2=MERIT2 + f - 1;
      end
    else
      MERIT2=mg+f;
    end
  end
%------------Finished Line Search------------

  if OPTIONS(7)~=5
    mf=abs(OPTIONS(18));
    LAMBDA=mf * LAMBDA+(1-mf) * LOLD;
  end
  % Test stopping conditions (convergence)
  if max(abs(SD))<2 * OPTIONS(2) & abs(gf' * SD)<2 * OPTIONS(3) & ...
        (mg<OPTIONS(4) | (strncmp(howqp,'i',1) & mg > 0 ) )
     if OPTIONS(1)>0
        if OPTIONS(7)==1,
          gamma=mg+f;
          disp([sprintf('%5.0f %12.6g  ',OPTIONS(10),gamma),...
                sprintf('%12.3g  ',OPTIONS(18)),how, '       ',howqp]);
        else
          disp([sprintf('%5.0f %12.6g %12.6g  ',OPTIONS(10),f,mg),...
                sprintf('%12.3g  ',OPTIONS(18)),how, '       ',howqp]);
        end
        if ~strncmp(howqp, 'i', 1)
          disp('Optimization Converged Successfully')
          active_const=find(LAMBDA>0);
          if active_const
            disp('Active Constraints:'),
            disp(active_const)
```

```
        else % active _ const == 0
          disp(' No Active Constraints');
        end
      end
    end
    if (strncmp(howqp, 'i',1) & mg > 0)
      disp('Warning: No feasible solution found.')
    end
    status=1;
  else % continue
    % NEED=[LAMBDA>0] | G>0
    if OPTIONS(10) >= OPTIONS(14)| OPT _ STOP
      XOUT=MATX;
      f=OLDF;
      if ~OPT _ STOP
        disp('Maximum number of function evaluations exceeded;')
        disp('increase OPTIONS(14)')
      end
      status=1;
    end
  end
end % while status ~= 1

% Update
OPTIONS(12)=OPTIONS(11);

% If a better unconstrained solution was found earlier, use it:
if f > bestf
  XOUT=bestx;
  f=bestf;
  HESS=bestHess;
  lambda=bestlambda;
end
OPTIONS(8)=f;
x(:)=XOUT;
if (OPT _ STOP)
  disp('Optimization terminated prematurely by user')
end
```

12. function

[x,OPTIONS,lambda,HESS]=constr(FUN,x,OPTIONS,VLB,VUB,GRADFUN,varargin)

函数 CONSTR 用于对一个多变量函数的约束寻优。

%

【例子】

(1) X=CONSTR('FUN',X0)

输入参数：

FUN 为一字符串参数，用以表示目标函数和约束条件，如：

funf='f=exp(x(1)) * (4 * x(1)^2 + 2 * x(2)^2 + 4 * x(1) * x(2) + 2 * x(2) + 1);';

fung='g=[1.5+x(1) * x(2)-x(1)-x(2);-x(1) * x(2)-10];';

FUN=[funf fung]；参见使用示例；

X0 为初始搜索坐标位置；

输出参数：X 为最终坐标位置；

(2) X=CONSTR('FUN',X,OPTIONS,VLB,VUB)

输入参数：

FUN 为一字符串参数，用以表示目标函数和约束条件，如：

funf='f=exp(x(1)) * (4 * x(1)^2 + 2 * x(2)^2 + 4 * x(1) * x(2) + 2 * x(2) + 1);';

fung='g=[1.5 + x(1) * x(2)-x(1)-x(2);-x(1) * x(2)-10];';

FUN=[funf fung]；参见使用示例；

X 为初始搜索坐标位置；

OPTIONS 为最优化计算工具箱通用选择参数，参见有关它的说明；

VLB 为搜索区间下界，如：VLB=zeros(1,2)，表示 X >= 0；

VUB 为搜索区间上界，如：VUB=[ ]，表示没有上界；

输出参数：X 为最终坐标位置：

(3) X=CONSTR('FUN',X,OPTIONS,VLB,VUB,'GRADFUN')

输入参数：

FUN 为一字符串参数，用以表示目标函数和约束条件，如：

funf='f=exp(x(1)) * (4 * x(1)^2 + 2 * x(2)^2 + 4 * x(1) * x(2) + 2 * x(2) + 1);';

fung='g=[1.5 + x(1) * x(2)-x(1)-x(2);-x(1) * x(2)-10];';

FUN=[funf fung]；参见使用示例；

X 为初始搜索坐标位置；

OPTIONS 为最优化计算工具箱通用选择参数，参见有关它的说明；

VLB 为搜索区间下界，如 VLB=zeros(1,2)，表示 X >= 0；

VUB 为搜索区间上界，如 VUB=[ ]，表示没有上界；

GRADFUN 为一字符串参数，用以表示目标函数和约束条件的导函数，如：

dfdx1='exp(x(1)) * (4 * x(1)^2+2 * x(2)^2+4 * x(1) * x(2)+2 * x(2)+1)+4 * exp(x(1)) * (2 * x(1)+x(2));';

dfdx2='4 * exp(x(1)) * (x(1)+x(2)+0.5);';

dfdg='[x(2)-1, -x(2); x(1)-1, -x(1)];';

GRADFUN=['df=[',dfdx1, dfdx2,']; dg=', dfdg];

输出参数：X 为最终坐标位置；

(4) [X,OPTIONS,LAMBDA,HESS]=
CONSTR('FUN',X,OPTIONS,VLB,VUB,'GRADFUN',P1,P2,…)

输入参数：

FUN 为一字符串参数，用以表示目标函数和约束条件；

X 为初始搜索坐标位置；

OPTIONS 为最优化计算工具箱通用选择参数，参见有关它的说明；

VLB 为搜索区间下界，如：VLB=zeros(1,2)，表示 X >= 0；

VUB 为搜索区间上界，如：VUB=[ ]，表示没有上界；

GRADFUN 为一字符串参数，用以表示目标函数和约束条件的导函数；

P1,P2,…:为以 M 文件表示的目标函数所需调用的参数。

输出参数:

X 为最优坐标位置;

OPTIONS 为最优化计算工具箱通用选择参数,参见有关它的说明

LAMBDA 为在最优坐标位置时的 Lagrange 乘子值;

HESS 为在最优坐标位置时的 Hessian 矩阵拟牛顿(quasi-Newton)近似值;

【程序】

```
if nargin < 2, error('constr requires two input arguments'); end
if nargin < 3, OPTIONS=[]; end
if nargin < 4, VLB=[]; end
if nargin < 5, VUB=[]; end
if nargin < 6, GRADFUN=[]; end

caller='constr';
lenVarIn=length(varargin);

% Convert to inline function as needed.
gradflag=0;
[funfcn,msg]=prefcnchk(FUN,caller,lenVarIn,gradflag);
if ~isempty(msg)
  error(msg);
end

if ~isempty(GRADFUN)
  gradflag=1;
  [gradfcn,msg]=prefcnchk(GRADFUN,caller,lenVarIn,gradflag);
  if ~isempty(msg)
    error(msg);
  end
else
  gradfcn=[];
end

[x,OPTIONS,lambda,HESS]=nlconst(funfcn,x,OPTIONS,VLB,VUB,gradfcn,varargin{:});
```

13. function[obj _ best,c _ best]=vsgsa(DimN,region,belta0,delta _ r,delta _ r _ limit);

函数 vsgsa 为变步长广义模拟退火算法子程序。

【例子】

function[obj _ best,c _ best]=vsgsa(DimN,region,belta0,delta _ r,delta _ r _ limit);

输入参数:

DimN 为优化问题的维数;

region 为优化区间,其中 region(:,1) 优化区间下限;region(:,2) 优化区间上限;

例如:region=[−5 6

−5 6

……

……

……];

delta _ r 为初始步长，根据具体优化问题选定，如：delta _ r=[0.15 0.15 …]'，在程序运行中将会自动调整

delta _ r _ limit 为终止步长，根据实际情况选定，如 delta _ r _ limit= 1 * 10^(-3)

belta0 为退火参数，凭经验选取。

%program for generalized simulated anneal (GSA) applied in CBBL

%According to Kalivas etc. AC 1991,63 ; GSA with variable step size

【程序】

```
b=region;

%  c _ c=[rand * b(1,2) rand * b(2,2)]';

current _ location=zeros(DimN,1);
for i=1:DimN
  current _ location(i)=rand * (region(i,2)-region(i,1))+region(i,1);
end

original _ location=current _ location;

obj _ c=obj _ fun(current _ location);%目标函数子程序

unfav _ count=0;
  not _ fall _ in=1;
succesive _ not _ accept=0;
  sample=0;
calcu0=zeros(5000,3);
  contract _ num=0;
c0=100 * obj _ c;
  obj _ best=10000;
  belta _ new=belta0;
count _ 11=1;
  count _ 22=1;
  change _ index=0;
%dec _ factor=.8;
  inc _ factor=1.01;
dec _ factor=.3;
  inc _ factor=1.005;
para0=0;
  para1=1;

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
while 1
    while not _ fall _ in
      u=randn(DimN,1);
      u=u/norm(u);
      c _ t=current _ location+u. * delta _ r;

      for i=1:DimN
        if c _ t(i)>=region(i,1) & c _ t(i)<=region(i,2)
          not _ fall _ in=0;
        else
          not _ fall _ in=1;
        end
      end

    end

    sample=sample+1;

    obj _ t=obj _ fun(c _ t);

    if obj _ t<=obj _ c
      obj _ c=obj _ t;
      current _ location=c _ t;
            calcu0(sample,3)=1; %accepted
      if obj _ c<obj _ best
        obj _ best=obj _ c;
        c _ best=current _ location; %%%%Best result during the session

      %succesive _ not _ accept=0;
     end
  else
     calcu0(sample,1)=1; %First row unfavorable number

        if count _11==1,
          differ0=obj _ t-obj _ c; count _11=count _11+1;
        elseif count _11==2,
          differ0=obj _ t-obj _ c+differ0; count _11=count _11+1;
        elseif count _11==3,
          differ0=obj _ t-obj _ c+differ0; count _11=count _11+1;
        end
        if count _11>3
          count _22=count _22+1;
          if count _22=-1
            change _ index=1;
```

```
          else
            change_index=0;
          end
        end
        if change_index==-1
          differ1=obj_t-obj_c;
          belta_new=belta0*differ0/(3*differ1);
          para0=sample; para1=1;
        end
      if exp(-belta_new*(obj_t-obj_c)/(obj_c-c0))>rand
        obj_c=obj_t;
        current_location=c_t;
        calcu0(sample,2)=1; %Second row acceptable unfavorable number
              calcu0(sample,3)=1;
        succesive_not_accept=0;

      else
         succesive_not_accept=succesive_not_accept+1;
      end
    end
obj_best
c_best

    max_delta_r=max(delta_r);
    if max_delta_r<delta_r_limit
           disp('Convergence of algorithm');
           t2=cputime;
           total_time=t2-t1;
           pause;
    end
    if sample<=100
         if (sample-para0)==20
           para0=sample;
           [sample sum(calcu0(sample-19:sample,2))/sum(calcu0(sample-19:sample,1)) current_location(1)
current_location(2) obj_c
norm(delta_r)]
         end
      else
         if (sample-para0)==20 & para1
           %[sample/100 sum(calcu0(sample-20:sample,1))
sum(calcu0(sample-20:sample,2)) sum(calcu0(sample-
20:sample,2))/sum(calcu0(sample-20:sample,1)) current_location(1)
current_location(2) obj_c succesive_not_accept norm(delta_r)
sum(calcu0(sample-19:sample,3))/20]
```

```
            if sum(calcu0(sample-19:sample,3))/20>0.9
            delta_r=delta_r * inc_factor;
            change_index=1; count_22=0; para1=0

        elseif sum(calcu0(sample-19:sample,3))/20<0.2
            delta_r=delta_r * dec_factor;
            change_index=1; count_22=0;para1=0;
        end

        para0=sample;
        [sample sum(calcu0(sample-19:sample,2))/sum(calcu0(sample-
19:sample,1)) current_location(1) current_location(2) obj_c
norm(delta_r)]
      end
    end
    not_fall_in=1; % To generate a new random vector;
    c0=obj_best/10;
end
var_opt=c;
```

14. function obj=obj_fun(location)

函数 obj_fun 为变步长广义模拟退火算法中调用的一个自定义的具体目标函数。

输入参数:location 为函数代入矢量,其形式为

x=location(1);
y=location(2);
…………………;
z=location(DimN);

输出参数:obj 为计算所得的目标函数值。

```
x=location(1);
y=location(2);

obj=2 * exp(-0.5 * ((x-1).^2+(y-3). * (y-3))) …
   +exp(-0.5 * ((x+2.5).^2+(y+5.5). * (y+5.5))) …
   +exp(-0.5 * ((x-2).^2+(y+2). * (y+2)));

obj=-obj;
```

15. function GLPoiX=glpoix(DimNum,PoiNum);

函数 GLPoiX 用于产生在(0,1)区间的好格子点。此程序可用于2或3维空间的好格子点,并将结果在图上显示。

【例子】

GLPoiX=glpoix(DimNum,PoiNum);

输入参数:

DimNum 为给定维数；

PoiNum 为给定点数，对于二维可给定以下数目：

8 13 21 34 55 89 144 233 377 610 987

对于三维可给定以下数目：

35 101 135 185 266 418 597 828 1010 1220 1459

输出参数：GLPoiX 为相应空间的好格子点。

参考文献：

KaiTai Fan, Yuan Wang

"Number-theoretic Methods in Statistic", Chapman & Hall, (1994).

【程序】

```
X=[8 13 21 34 55 89 144 233 377 610 987
1 1 1 1 1 1 1 1 1 1 1
5 8 13 21 34 55 89 144 233 377 610];
X1=[35 101 135 185 266 418 597 828 1010 1220 1459
1 1 1 1 1 1 1 1 1 1 1
11 40 29 26 27 90 63 285 140 319 256
16 85 42 64 69 130 169 358 237 510 373];

if DimNum==3
X=X1;
end
GLPoiX=zeros(DimNum,PoiNum);
[m,n]=size(X);
  for i=1:n
   if X(1,i)==PoiNum
   k=i;
   else
   end
  end

  for i=1:DimNum
   for j=1:PoiNum
   GLPoiX(i,j)=(j*X(i+1,k)-0.5)/X(1,k)-fix((j*X(i-1,k)-0.5)/X(1,k));
   end
  end
  if DimNum==2
    plot(GLPoiX(1,:),GLPoiX(2,:),'.')
  end

   if DimNum==3
     subplot(221),plot(GLPoiX(1,:),GLPoiX(2,:),'.')
     subplot(222),plot(GLPoiX(1,:),GLPoiX(3,:),'.')
     subplot(223),plot(GLPoiX(2,:),GLPoiX(3,:),'.')
end
```

16. function [opti,reg]=sntopti(b,DimN,PoiN,y,x1);

函数 sntopti 为序贯数论优化子程序。

【例子】

[opti,reg]=sntopti(b,DimN,PoiN,y,x1);

输入参数：

DimNum 为给定维数；

PoiNum 为给定点数，对于二维可给定以下数目：8 13 21 34 55 89 144 233 377 610 987 对于三维可给定以下数目：

35 101 135 185 266 418 597 828 1010 1220 1459

b 为待优化的初始矢量；

y,x1为待优化函数中的所需数据，本例中为相应矢量和校正矩阵。

输出参数：

opti 所得最优目标函数值；

reg 为相应的空间位置。

【程序】

```
respOpt=5000.;
GLPoiX=glpoix(DimN,PoiN);
size(GLPoiX)
pause
for k=1:10

    for i=1:DimN
    b1(i,1)=(b(i,2)-b(i,1));
    end
    for j=1:PoiN
       resp=lmsobj(b1.*GLPoiX(:,j)+b(:,1),y,x1);

         if resp<=respOpt
         respOpt=resp
      OptPoint(:,1)=b1.*GLPoiX(:,j)+b(:,1)
      else
      end
    end
    opti=respOpt;
    reg=OptPoint;

    for i=1:DimN
    btem1=[OptPoint(i,1)-b1(i,1)/4 b(i,1)];
    btem2=[OptPoint(i,1)+b1(i,1)/4 b(i,2)];
    b(i,1)=max(btem1);
    b(i,2)=min(btem2);
    end
    disp(b)
```

```
end
end
```

17. function son=mate(popu)

函数 mate 为遗传算法中的交叉函数。

【例子】

son=mate(popu)

输入参数:popu 为染色体。

输出参数:son 为交配过程中产生出来的子代。

【程序】

```
[dimen,psize]=size(popu);
son=zeros(dimen,psize);
matesite=round(rand(dimen,psize/2)).*(rand(dimen,psize/2)-0.5)*2;
j=0;
for i=1:2:psize
    j=j+1;
    son(:,i)=popu(:,i)+(popu(:,i+1)-popu(:,i)).*matesite(:,j);
    son(:,i+1)=popu(:,i+1)-(popu(:,i+1)-popu(:,i)).*matesite(:,j);
end
```

18. function son=mutate(popu,pmutate,mutatec)

函数 mutate 为遗传算法中的变异函数。

【例子】

son=mutate(popu,pmutate,mutatec)

输入参数:

popu 为染色体;

pmutate 为遗传进化中的变异率;

mutatec 为变异步长 。

输出参数:son 为变异过程中产生出来得子代。

【程序】

```
site=rand(size(popu))<pmutate;
son=popu+site.*(rand(size(popu))-0.5)*2*mutatec;
```

19. function parent=select(popufit)

函数 select 为遗传算法中的选择性交叉函数。

【例子】

parent=select(popufit)

输入参数:popufit 为染色体适应性。

输出参数:parent 选作为父代的染色体。

【程序】

```
psize=length(popufit);
fitsum=sum(popufit);
ind=rand(1,psize)*fitsum;
parent=ones(1,psize);
for i=1:psize
    parent=parent+(sum(popufit(1:i))<ind);
end
```

20. function offspr=compet(popufit,psize)

函数 compet 为遗传算法中的竞争函数

【例子】

offspr=compet(popufit,psize)

输入参数：

popufit 为染色体适应性；

psize 染色体个体数目

输出参数:offspr 通过竞争产生的新子代。

【程序】

```
offspr=zeros(1,psize);
[mixima,ind]=max(popufit);
offspr(1)=ind;popufit(ind)=0;
for i=2:psize
    fitsum=sum(popufit);
    ind=rand * fitsum;
    j=1;
    while sum(popufit(1:j))<ind
      j=j+1;
    end
    offspr(i)=j;popufit(j)=0;
end
```

21. function popufit=fitness(popuerr)

函数 fitness 为遗传算法中用于计算适应性的子程序；

【例子】

popufit=fitness(popuerr)

输入参数:popuerr 与染色体对应的误差。

输出参数:popufit 为染色体适应性。

【程序】

```
popufit=-ones(size(popuerr))./popuerr;
```

22. function H = hadamard(n)

函数 HADAMARD 用于产生 Hadamard 矩阵。

【例子】

H=hadamard(N)

输入参数:N 表示想要得到的 Hadamard 矩阵的阶数。

输出参数:H 为一只含元素1或-1的 Hadamard 矩阵，即 H' * H = N * EYE(N)。

注意:n 为一个可被4整除的数，而且，此程序只能用于当 N，N/12 or N/20都是2的幂函数的情况，即 $N=2^k$;$N/12=2^t$;$N/20=2^t$ 的情况。

【程序】

```
[f,e]=log2([n n/12 n/20]);
k=find(f==1/2 & e>0);
if min(size(n)) > 1 | isempty(k)
   error(['n must be an integer and n,n/12 or n/20 must be a power of 2.']);
end
e=e(k)-1;
```

```
if k==1      % N=1 * 2^e;   H=1;

elseif k==2      % N=12 * 2^e;
   H=[ones(1,12);ones(11,1) ...
       toeplitz([-1 -1 1 -1 -1 -1 1 1 1 -1 1],[-1 1 -1 1 1 1 -1 -1 -1 1 -1])];

elseif k==3      % N=20 * 2^e;
   H=[ones(1,20);ones(19,1) ...
       hankel([-1 -1 1 1 -1 -1 -1 -1 1 -1 1 -1 1 1 1 1 -1 -1 1],... [1 -1 -1 1 1 -1 -1 -1
             -1 1 -1 1 -1 1 1 1 1 -1 -1])];
end

%  Kronecker product construction.
for i=1:e
   H=[H  H
      H  H];
end
```

23. function [b,bint,r,rint]=regress(y,X,alpha)

函数 REGRESS 用最小二乘方法来进行多元线性回归。

【例子】

(1) b=REGRESS(y,X)

输入参数：

X 是一个(n×p)阶校正矩阵；

y 是一个(n×1)阶相应矢量。

输出参数：b 是一个(p×1)回归矢量，即 y=Xb。

(2) [B,BINT,R,RINT,STATS]=REGRESS(y,X,alpha)

输入参数：

X 是一个(n×p)阶校正矩阵；

y 是一个(n×1)阶相应矢量；

alpha 为置信水平的百分数；

输出参数：b 是一个(p×1)回归矢量，即 y=Xb；

BINT 为回归矢量的置信区间；

R 为残差矢量；

RINT 为残差矢量的置信区间；

参考文献：

1 Samprit Chatterjee and Ali S. Hadi, "Influential Observations, High Leverage Points, and Outliers in Linear Regression", Statistical Science 1986 Vol. 1 No. 3 pp. 379~416

2 N. Draper and H. Smith, "Applied Regression Analysis, Second Edition", Wiley, 1981

【程序】

```
if nargin < 2,
    error('REGRESS requires at least two input arguments.');
end
```

```
if nargin==2,
    alpha=0.05;
end

% Check that matrix (X) and left hand side (y) have compatible dimensions
[n,p]=size(X);
[n1,collhs]=size(y);
if n ~=n1,
    error('The number of rows in Y must equal the number of rows in X.');
end

if collhs ~=1,
    error('Y must be a vector,not a matrix');
end

% Find the least squares solution.
[Q,R]=qr(X,0);
b=R\(Q' * y).

% Find a confidence interval for each component of x
% Draper and Smith,equation 2.6.15,page 94

RI=R\eye(p);
xdiag=sqrt(sum((RI .* RI)'))';
nu=n-p;                 % Regression degrees of freedom.
yhat=X * b;             % Predicted responses at each data point.
r=y-yhat;               % Residuals.
if nu ~-0
   rmse=norm(r)/sqrt(nu);          % Root mean square error.
else
   rmse=Inf;
end
s2=rmse^2;              % Estimator of error variance.
tval=tinv((1-alpha/2),nu);
bint=[b-tval * xdiag * rmse,b+tval * xdiag * rmse];

% Calculate R-squared.
if nargout==5,
   RSS=norm(yhat-mean(y))^2;      % Residual sum of squares.
   TSS=norm(y-mean(y))^2;         % Total sum of squares.
   r2=RSS/TSS;                    % R-square statistic.
   F=(RSS/(p-1))/s2;              % F statistic for regression.
   proh=1-fcdf(F,p-1,nu);         % Significance probability for regression
   stats=[r2 F prob];
```

```
end

% Find the standard errors of the residuals.
% Get the diagonal elements of the "Hat" matrix.
% Calculate the variance estimate obtained by removing each case (i.e. sigmai).
% see Chatterjee and Hadi p. 380 equation 14.
T=X * RI;
hatdiag=sum((T .* T)')';
if nu<1,
  ser=rmse * ones(length(y),1);
elseif nu>1
  sigmai=sqrt((nu * s2/(nu-1)) - (r .^2 ./ ((nu-1) .* (1-hatdiag))));
  ser=sqrt(1-hatdiag) .* sigmai;
elseif nu==1
  ser=sqrt(1-hatdiag) .* rmse;
end

ti=r ./ ser;

% Create confidence intervals for residuals.
Z=[(r-tval * ser) (r+tval * ser)]';
rint=Z';
```

24. function regstats(responses,data,model)

函数 REGSTATS 用于对线性模型的回归诊断。该函数提供了一个对话界面，可得几乎所有单点诊断的信息。

【例子】

REGSTATS(RESPONSES,DATA,MODEL)

输入参数：

RESPONSES 为一(n×1)阶相应矢量；

DATA 是一个(n×p)阶校正矩阵；

字符串参数 MODEL 表示所选择的回归模型，其预置值为一个包括常数项的线性加合模型；

当 MODEL= 'interaction'时，表示一个包括常数项，线性项和交叉项的回归模型；

当 MODEL= 'quadratic'时，表示一个包括常数项、线性项、交叉项以及平方项的回归模型；

当 MODEL= 'purequadratic'时，表示一个包括常数项、线性项和平方项的回归模型。

输出参数：

Q　从 QR 分解中得到的 Q；

R　从 QR 分解中得到的 R；

beta　回归系数；

covb　回归系数的方差；

yhat　相应矢量的拟合值；

r　残差矢量；

mse　均方差；

Leverage　强影响点矢量；

Hatmat　帽子矩阵；

s2_i 剔除 i 点数据后的方差；

beta _i 剔除 i 点数据后的回归系数；

Standres 标准化残差；

studres 学生化残差；

dfbeta 回归系数的变化；

dffit 拟合矢量值的变化；

dffits 拟合矢量值标度变化;Scaled Change in Fitted Values.'

Covratio 协方差变化率；

cookd Cook 距离。

【程序】

```
if ~isstr(responses)
    action='start';
else
    action=responses;
end

%  On recursive calls get all necessary handles and data.
if ~strcmp(action,'start')
  diagfig=findobj('Tag','diagfig');
  ud=get(diagfig,'UserData');
  y=ud{1};
  X=ud{2};
  idx=ud{3};
  texthandle=ud{4};
end

%  Initialize GUI
if strcmp(action,'start')
  if nargin < 3
    model='linear';
  end
  X=x2fx(data,model);
  diagfig=figure('Units','Normalized',...
       'Position',[0.1 0.5 0.5 0.5],'Color',[.5 .5 .5],...
    'Userdata',{responses,X,[],[]},'Tag','diagfig');

  textlables=str2mat('Q from QR Decomposition','R from QR Decomposition','Coefficients','Coefficient
Covariance','Fitted Values');
  textlables=str2mat(textlables,'Residuals','Mean Square Error');
  textlables=str2mat(textlables,'Leverage','Hat Matrix','Delete-1 Variance');
  textlables=str2mat(textlables,'Delete-1
Coefficients','Standardized Residuals');
  textlables=str2mat(textlables,'Studentized Residuals','Change in Beta',...
              'Change in Fitted Value','Scaled change in Fit');
```

```
textlables=str2mat(textlables,'Change in Covariance','Cook' 's Distance');

varnames={'Q','R','beta','covb','yhat','r','mse','leverage','hatmat','s2_i',...
'beta _i','standres','studres','dfbeta','dffit','dffits','covratio','cookd'};

uicontrol(diagfig,'Style','text','Units','normalized',...
     'Position',[0.02 0.92 0.4 0.04],...
     'BackgroundColor',[.5 .5 .5],...
     'String','Regression Diagnostics','ForegroundColor','k');
   uicontrol(diagfig,'Style','Pushbutton','Units','normalized',...
       'Position',[0.5 0.92 0.3 0.06],...
       'BackgroundColor',[.5 .5 .5],...
       'String','Calculate Now','ForegroundColor','k',...
       'Callback',['regstats("calculate")']);
   chkboxes=zeros(18,1);
    for k=1:18
       boxcol=rem(k-1,9)+1;
       boxrow=floor(k/9-eps);
       chkboxes(k)=
uicontrol(diagfig,'Style','checkbox','Units','normalized',...
     'Position',[0.02+0.45*boxrow 0.9-0.1*boxcol 0.4 0.07],....
     'BackgroundColor','w','String',textlables(k,:),'Tag',int2str(k),...
     'UserData',varnames{k},'ForegroundColor','k');
   end

close _ button =uicontrol('Style','Pushbutton','Units','normalized',...
            'Position',[0.90 0 0.07
0.07],'Callback','close','String','Close');

help _ button =uicontrol('Style','Pushbutton','Units','normalized',...
             'Position',[0.90 0.10 0.07
0.07],'Callback','regstats("help")','String','Help');

elseif strcmp(action,'calculate')
  varhandles=zeros(18,1);
  idx=(1:18)';
  varnames=cell(18,1);
  for k=18:-1:1
    varhandles(k)=findobj(gcf,'Tag',int2str(k));
    if get(varhandles(k),'Value')==0,idx(k)=[];end
    varnames{k}=get(varhandles(k),'UserData');
  end

  for k=1:length(idx)
```

```
    if k==1
      entrystr=varnames{idx(k)};
    else
      entrystr=[entrystr,',',varnames{idx(k)}];
    end
  end
  instr=varnames(idx);
  promptstr='Enter variable names for regression diagnostics or use defaults.';
  dlghandle=dialog('Visible','off','WindowStyle','modal');

  set(dlghandle,'Units','pixels','Position',[100 300 400
100],'Visible','on');
  uicontrol(dlghandle,'Style','text','Units','pixels',...
        'Position',[10 75 380 20],...
        'String',promptstr);

  texthandle =uicontrol(dlghandle,'Style','edit','Units','pixels',...
            'String',entrystr,'HorizontalAlignment','left');
  txtlength=get(texthandle,'Extent');
  if txtlength(3) > 350
    pos=[10 30 380 40];
    set(texthandle,'Max',2,'Position',pos)
  else
    pos=[(180 - (ceil(txtlength(3)/2))+15) 40 txtlength(3)+29 20];
    set(texthandle,'Position',pos)
  end

  ok_button =uicontrol('Style','Pushbutton','Units','pixels',...
            'Position',[180 5 40
20],'Callback','regstats(''output'');close(gcbf)','String','OK');

  ud{3}=idx;
  ud{4}=texthandle;
  set(diagfig,'UserData',ud);
elseif strcmp(action,'output')
  [Q,R]=qr(X,0);
  beta=R\(Q' * y);
  yhat=X * beta;
  residuals=y-yhat;
  nobs=length(y);
  p=min(size(R));
  mse=sum(residuals. * residuals). /(nobs-p);
  E=X/R;
  h=sum((E. * E)')';
```

```
s_sqr_i=((nobs-p)*mse - residuals.*residuals./(1-h))./(nobs-p-1);
e_i=residuals./sqrt(s_sqr_i.*(1-h));
ri=R\eye(p);
xtxi=ri*ri';
s=get(texthandle,'String');
cpos=findstr(s,',');
if isempty(cpos)
  C={s};
else
  nvars=length(cpos)+1;
  C=cell(nvars,1);
  for k=1:nvars
      if k==1
         C{k}=s(1:cpos(1)-1);
      elseif k==nvars
         C{k}=s((cpos(k-1)+1):end);
      else
         C{k}=s((cpos(k-1)+1):(cpos(k)-1));
      end
  end
end

if (any(idx==1))
   % Q from the QR decomposition of the X matrix.
   Qstr=C{find(idx==1)};
   Qstr=fliplr(deblank(fliplr(Qstr)));
   assignin('base',Qstr,Q);
end
if (any(idx==2))
   % R from the QR decomposition of the X matrix.
   Rstr=C{find(idx==2)};
   Rstr=fliplr(deblank(fliplr(Rstr)));
   assignin('base',Rstr,R);
end
if (any(idx==3))
   % Coefficients.
   bstr=C{find(idx==3)};
   bstr=fliplr(deblank(fliplr(bstr)));
   assignin('base',bstr,beta);
end
if (any(idx==4))
   % Covariance of the parameters.
   covb=xtxi*mse;
   covbstr=C{find(idx==4)};
```

```
    covbstr=fliplr(deblank(fliplr(covbstr)));
    assignin('base',covbstr,covb);
end
if (any(idx==5))
    % Fitted values.
    yhatstr=C{find(idx==5)};
    yhatstr=fliplr(deblank(fliplr(yhatstr)));
    assignin('base',yhatstr,yhat);
end
if (any(idx==6))
    % Residuals.
    rstr=C{find(idx==6)};
    rstr=fliplr(deblank(fliplr(rstr)));
    assignin('base',rstr,residuals);
end
if (any(idx==7))
    % Mean squared error.
    msestr=C{find(idx==7)};
    msestr=fliplr(deblank(fliplr(msestr)));
    assignin('base',msestr,mse);
end
if (any(idx==8))
    % Leverage.
    hstr=C{find(idx==8)};
    hstr=fliplr(deblank(fliplr(hstr)));
    assignin('base',hstr,h);
end
if (any(idx==9))
    % Hat Matrix.
    hatstr=C{find(idx==9)};
    hatstr=fliplr(deblank(fliplr(hatstr)));
    assignin('base',hatstr,Q*Q');
end
if (any(idx==10))
    % Delete 1 variance. S_I
    dlvstr=C{find(idx==10)};
    dlvstr=fliplr(deblank(fliplr(dlvstr)));
    assignin('base',dlvstr,s_sqr_i);
end
if (any(idx==11)) | (any(idx==14))
    % Delete 1 coefficients. BETA_I
    stde=residuals./(1-h);
    stde=stde(:,ones(p,1));
    b_i=beta(:,ones(nobs,1)) - ri*(Q.*stde)';
```

```
    if (any(idx==11)),
        dlbstr=C{find(idx==11)};
        dlbstr=fliplr(deblank(fliplr(dlbstr)));
        assignin('base',dlbstr,b_i);
    end
end
if (any(idx==12))
    % Standardized residuals.
    rlstr=C{find(idx==12)};
    rlstr=fliplr(deblank(fliplr(rlstr)));
    assignin('base',rlstr,residuals./(sqrt(mse*(1-h))));
end
if (any(idx==13))
    % Studentized residuals.
    r2str=C{find(idx==13)};
    r2str=fliplr(deblank(fliplr(r2str)));
    assignin('base',r2str,e_i);
end
if (any(idx==14))
    % Change in beta. DFBETAS
    b=beta(:,ones(nobs,1));
    s=sqrt(s_sqr_i(:,ones(p,1))');
    rtri=sqrt(diag(xtxi));
    rtri=rtri(:,ones(nobs,1));
    dfbeta=(b - b_i)./(s.*rtri);
    dfbstr=C{find(idx==14)};
    dfbstr=fliplr(deblank(fliplr(dfbstr)));
    assignin('base',dfbstr,dfbata);
end
if (any(idx==15))
    % Change in fitted values. DFFIT
    dffitstr=C{find(idx==15)};
    dffitstr=fliplr(deblank(fliplr(dffitstr)));
    assignin('base',dffitstr,h.*residuals./(1-h));
end
if (any(idx==16))
    % Scaled change in fitted values. DFFITS
    sdffitstr=C{find(idx==16)};
    sdffitstr=fliplr(deblank(fliplr(sdffitstr)));
    assignin('base',sdffitstr,sqrt(h./(1-h)).*e_i);
end
if (any(idx==17))
    % Change in covariance. COVRATIO
    covr=(((nobs-p-1+e_i.*e_i)./(nobs-p)).^p)./(1-h);
```

```
    covrstr=C{find(idx==17)};
    covrstr=fliplr(deblank(fliplr(covrstr)));
    assignin('base',covrstr,covr);
  end
  if (any(idx==18))
    %   Cook's Distance.
    d=residuals. * residuals. * (h. /(1-h)). /p;
    cookstr=C{find(idx==18)};
    cookstr=fliplr(deblank(fliplr(cookstr)));
    assignin('base',cookstr,d);
  end

elseif strcmp(action,'help')
str{1,1}='Regression Diagnostics';
str{1,2}={
'Regression Diagnostics'
'        '
'The literature suggests many diagnostic statistics for evaluating'
'multiple linear regression.  The usual regression model is:'
'        '
'y=Xb+e'
'where y is an n by 1 vector of responses,'
'       x is an n by p matrix of predictors,'
'       b is an p by 1 vector of parameters,'
'       e is an n by 1 vector of random disturbances.'
'        '
'Let X=Q * R where Q and R come from a QR Decomposition of X.'
'Q is orthogonal and R is triangular.  Both these matrices are'
'useful for calculating many regression diagnostics.'
'        '
'The least squares estimator for b is:'
'b=R\(Q'        ' * y);'
'        '
'To learn more about the diagnostics supplied by regstats,use the'
'Topics popup menu.  A list of the Topics is below: '
'   □Q from QR Decomposition.'
'   □R from QR Decomposition.'
'   □Regression Coefficients.'
'   □Covariance of regression coefficients.'
'   □Fitted values of the response data.'
'   □Residuals.'
'   □Mean Squared Error.'
'   □Leverage  '
'   □Hat Matrix.'
```

```
'  □Delete-1 Variance.'
'  □Delete-1 Coefficients.'
'  □Standardized Residuals.'
'  □Studentized Residuals.'
'  □Change in Regression Coefficients.'
'  □Change in Fitted Values.'
'  □Scaled Change in Fitted Values.'
'  □Change in Covariance.'
'  □Cook'     's Distance.'
};
str{2,1}='QR Decomposition (Q)';
str{2,2}={
'Q from the QR Decomposition of X.'
'     '
'MATLAB code:'
'[Q,R]=qr(X,0);'
'     '
'This is the so-called economy sized QR decomposition.'
'Q is n by p. Q is also orthogonal. That is:'
'Q'     '*Q=I (the identity matrix)'
};

str{3,1}='QR Decomposition (R)';
str{3,2}={
'R from the QR Decomposition of X.'
'     '
'MATLAB code:'
'[Q,R]=qr(X,0);'
'     '
'This is the economy sized QR decomposition.'
'R is p by p and triangular. This makes solving linear '
'systems simple.'
};

str{4,1}='Regression Coefficients';
str{4,2}={
'Regression Coefficients'
'     '
'MATLAB code:'
'b=R\(Q'     '*y);'
'     '
'If you only wanted the coefficients and '
'did not want to use Q and R later,then:'
'b=X\y;'
```

```
'is the simplest code.'
};

str{5,1}='Coefficient Covariance';
str{5,2}={
'Covariance of Regression Coefficients'
'   '
'MATLAB code:'
'%Inverse of R'
'ri=R\eye(p);'
'%inv(X'' * X)'
'xtxi=ri * ri'';'
'%Covariance of the coefficients.'
'covb=xtxi * mse;'
'   '
'MSE is mean squared error.'
'   '
'covb is a p by p matrix. The diagonal elements are the variances'
'of the individual coefficients.'
};

str{6,1}='Fitted Values';
str{6,2}={
'Fitted Values of the Response Data'
'   '
'The usual regression model is:'
'   '
'y=Xb+e'
'where y is an n by 1 vector of responses,'
'      x is an n by p matrix of predictors.'
'      b is an p by 1 vector of parameters,'
'      e is an n by 1 vector of random disturbances.'
'   '
'Let X=Q * R where Q and R come from a QR Decomposition of X.'
'Q is orthogonal and R is triangular. Both these matrices are'
'useful for calculating many regression diagnostics.'
'   '
'The least squares estimator for b is:'
'b=R\(Q'' * y);'
'   '
'Plugging this estimate for b into the model equation (leaving out e) we have:'
'yhat=X * b=X * (R\(Q'' * y))'
'where yhat is an n by 1 vector of fitted (or predicted) values of y.'
};
```

```
str{7,1}='Residuals';
str{7,2}={
'Residuals'
'  '
'Let y be an n by 1 vector of responses,and'
'yhat be an n by 1 vector of fitted (or predicted) values of y.'
'  '
'Then the residuals,also n by 1 are:'
'r=y-yhat'
'  '
'In English,the residuals are the observed values minus the predicted values.'
};

str{8,1}='Mean Squared Error';
str{8,2}=
'Mean Squared Error'
'  '
'The mean squared error is an estimator of the variance of the random'
'disturbances,which is assumed constant.'
'  '
'The formula is:'
'mse=r'          '*r./(n-p)'
'where r is the n by 1 vector of residuals.'
'          n is the number of observations.'
'          p is the number of unknown parameters.'
};

str{9,1}='Leverage';
str{9,2}={
'Leverage'
'  '
'The leverage is a measure of the effect of a particular observation on the'
'regression predictions due to the position of that observation in the space'
'of the inputs.'
'  '
'In general,the farther a point is from the center of the input space,the'
'more leverage it has.'
'  '
'The leverage vector is an n by 1 vector of the leverages of all the observations.'
'It is the diagonal of the hat matrix.'
'  '
'MATLAB code'
'h=diag(Q*Q'          ');'
```

```
};

str{10,1}='Hat Matrix';
str{10,2}={
'Hat (Projection) Matrix'
'     '
'The hat matrix is an n by n matrix that projects the vector of '
'observations,y,onto the vector of predictions,yhat,thus putting '
'the "hat" on y.'
'     '
'MATLAB code.'
'H      =Q * Q'        ';'
'yhat=H * y;'
};

str{11,1}='Delete-1 Variance';
str{11,2}={
'Delete-1 Variance'
'     '
'The delete-1 variance is an n by 1 vector. Each element of the vector is the'
'mean squared error of the regression obtained by deleting that observation.'
};

str{12,1}='Delete-1 Coefficients';
str{12,2}={
'Delete-1 Coefficients'
'     '
'The delete-1 coefficents a an p by n matrix. Each column of the matrix is the '
'coefficients of the regression obtained by deleting corresponding observation.'
};

str{13,1}='Standardized Residuals';
str{13,2}={
'Standardized Residuals'
'     '
'The standardized residuals are the residuals normalized by a measure of their'
'standard deviation.'
'     '
'MATLAB code.'
'stanres=residuals./(sqrt(mse * (1-h)));'
'where mse is the mean squared error.'
'          and h is the leverage vector.'
'     '
'See also: Studentized Residuals .'
```

```
};

str{14,1}='Studentized Residuals';
str{14,2}={
'Studentized Residuals'
'  '
'The studentized residuals are the residuals normalized by a measure of their '
'standard deviation.'
'  '
'MATLAB code:'
'studres=residuals./(sqrt(s2i*(1-h)));'
'where s2i is the delete-1 variance.'
'      and h is the leverage vector.'
'  '
'See also: Standardized Residuals.'
};

str{15,1}='Change in Coefficients';
str{15,2}={
'Change in Regression Coefficients (DFBETA)'
'  '
'DFBETA is an p by n matrix. Each column of DFBETA is the scaled effect of '
'removing the corresponding point on the vector of coefficients.'
};

str{16,1}='Change in Prediction';
str{16,2}={
'Change in Fitted Values (DFFIT)'
'  '
'DFFIT is an n by 1 vector. Each element is the change in the fitted value'
'caused by deleting the corresponding observation.'
'  '
'MATLAB code:'
'dffit=h.*residuals./(1-h)'
'where h is the leverage vector.'
};

str{17,1}='Scaled Change in yhat';
str{17,2}={
'Scaled Change in Fitted Values (DFFITS)'
'Scaled DFFIT is an n by 1 vector. Each element is the change in the fitted value '
'caused by deleting the corresponding observation and scaled by the standard error.'
'  '
'MATLAB code:'
```

```
'sdffit=sqrt(h./(1-h)).*e_i)'
'where e_i is the vector of studentized residuals.'
'        and h is the leverage vector.'
};

str{18,1}='Change in Covariance';
str{18,2}={
'Change in Covariance'
'      '
'DFCOV is an n by 1 vector. Each element is the ratio of the generalized '
'variance of the estimated coefficients when the corresponding element is '
'deleted to the generalized variance of the coefficients using all the data.'
};

str{19,1}='Cook''s Distance';
str{19,2}={
'Cook''s Distance'
'      '
'Cook''s distance is an n by 1 vector. Each element is the normalized change '
'in the vector of coefficients due to the deletion of an observation.'
'      '
'MATLAB code:'
'd=residuals.*residuals.*(h./(1-h))./p;'
'where d is Cook''s distance.'
'      h is the leverage vector.'
'      and p is the number of unknown parameters.'
};

  for k=1:19
     str{k,2}=char(str{k,2});
  end
  helpwin(str,'Regression Diagnostics','Regression Diagnostics Help')

end
```

25. function [cc,c,v]=kalmfil(x,y);

函数 kalmfil 可用于直接校正的 kalmam 估计

【例子】

[cc,c,v]=kalmfil(x,y);

输入参数:

x 是一个(n×p)阶校正矩阵;

y 为一(n×1)阶相应矢量。

输出参数:

c Kalman 递推估计解系列矩阵;

v 其信息系列矢量;

cc 最终估计矢量。

说明：该函数可用下述模拟例子来进行检验：

```
>[x1,y1]=simuld(3,[1 1 1],[25 35 45],[7 7 8],[.5 .6 .7],[35 43 50],[8 7.6 6],.04,[.5 .7 .6]');
```

此函数调用将产生一个模拟的3组分分析体系，其真实浓度为：[.5 .7 .6]，参见

simuld>[cc,c,v]=kalmfil(x,y);

用此函数调用可得 Kalman 递推估计解系列 c，其信息系列 v，及最终估计 cc。

```
% Check that matrix (x) and left hand side (y) have compatible dimensions
[n,p]=size(x);

[n1,collhs]=size(y);
if n~=n1,
    error('The number of rows in Y must equal the number of rows in X.');
end

if collhs ~=1,
    error('Y must be a column vector.');
end
[m,n]=size(x);
% initialize p matrix and etimates c
x1=1.2;
r=.000001;
c(:,1)=[0 0 0]';
I=eye(3,3);
p=190*sqrt(r./(x1*x1'))*I
% p=I
% pause

% recursive procedure
for i=1:m
xi=x(i,:)';
g=p*xi*(1/(xi'*p*xi+r));
v(i)=y(i)-xi'*c(:,i);
c(:,i+1)=c(:,i)+g*(v(i));
p=(I-g*xi')*p*(I-g*xi')'+g*r*g';
end
cc=c
plot(v)
```

26. function [rig]=ridgecal(x,y);

函数 ridgecal 可用于岭回归的岭迹计算来估计病态分析体系的回归系数。

【例子】

```
[rig]=ridgecal(x,y);
```

输入参数：

x 是一个(n×p)阶校正矩阵；

y 为一(n×1)阶相应矢量。

输出参数：rig 为计算所得的岭迹，其中进入平稳状态时的回归系数即为所得结果。

说明：该函数可用下述模拟例子来进行检验：

〉[x1,y1]=simuld(3,[1 1 1],[25 25 35],[7 7 8],[.5 .6 .7],[35 36 50],[8 7.6 6],.04,[.5 .7 .6]')；

此函数调用将产生一个模拟的3组分病态分析体系，其真实浓度为：[.5 .7 .6]，可其最小二乘估计为：[0.2686 0.9485 0.5719]. 参见 simuld

〉[rig]=ridgecal(x1,y1)；

用此函数调用可得一岭迹图，同时可得其进入平稳状态时的回归系数为：[0.5178 0.6921 0.5911]，已很靠近真实解。

【程序】

```
% Check that matrix (x) and left hand side (y) have compatible dimensions
[n,p]=size(x);

[n1,collhs]=size(y);
if n~=n1,
    error('The number of rows in Y must equal the number of rows in X.');
end

if collhs ~=1,
    error('Y must be a column vector.');
end

[m,n]=size(x);
[s,v,d]=svd(x);
startpoint=v(n,n)/200
endpoint=v(n,n)
(endpoint-startpoint)/100
pause
clear s d
for i=1:101
   k=startpoint * i;
   rig(:,i)=inv(x' * x+k * eye([n,n])) * x' * y;
end
plot([startpoint:(endpoint-startpoint)/100:endpoint],rig);
```

27. function [x,dx]=lscov(A,b,V)

函数 LSCOV 可用于直接校正的加权最小二乘估计.

【例子】

(1) X=LSCOV (A, B, V)

输入参数：

A 是一个（n×p）阶校正矩阵；

B 为一（n×1）阶相应矢量或（n×d）；

V 为量测误差的协方差矩阵，通常为对角矩阵（n×n）；

输出参数：X 为计算所得的加权最小二乘估计矢量或矩阵（p×d）由下式给出

X=inv（A′＊inv（V）＊A）＊A′＊inv（V）＊B

(2) [X,DX]=LSCOV(A,B,V)

输入参数意义同前；

输出参数：X 为计算所得的加权最小二乘估计矢量或矩阵(p×d),DX 由下述式子给出

mse=B′＊(inv(V)－inv(V)＊A＊inv(A′＊inv(V)＊A)＊A′＊inv(V))＊B./(m－n)

DX=sqrt(diag(inv(A′＊inv(V)＊A)＊mse))

说明:该函数可用下述模拟例子来进行检验:

参见 SLASH,NNLS,QR.

参考文献：

1 G. Strang,"Introduction to Applied Mathematics",Wellesley-Cambridge,p. 398,1986

2 F. Greybill,"Theory and Application of the Linear Model",Duxbury Press p. 207,1976

【程序】

```
[m,n]=size(A);

% Error checking
if m <=n,error('Problem must be over-determined so that M > N.');end
if size(b,1)~=m,
  error(sprintf('B must be a column vector of size %d-by-1.',m));
end
if ~isequal(size(V),[m m]),
  error(sprintf('V must be a %d-by-%d matrix.',m,m));
end

[qnull,r]=qr(A);
q=qnull(:,1:n);
r=r(1:n,1:n);
qnull(:,1:n)=[];

g=qnull'*V*qnull;
f=q'*V*qnull;

c=q'*b;
d=qnull'*b;

x=r\(c-f*(g\d));

if nargout==2
   % Standard formulas are:
```

```
  % mse=b' * (inv(V) - inv(V) * A * inv(A' * inv(V) * A) * A' * inv(V)) * b./(m-n);
  % dx=sqrt(diag(inv(A' * inv(V) * A) * mse));
  U=(chol(V))';
  z=U\b;
  W=U\A;
  mse=(z' * z - x' * W' * z)/(m-n);
  [Q,R]=qr(W,0);
  ri=(R\eye(n))';
  dx=(sqrt(sum(ri. * ri) * mse))';
end
```

28. function [p,cun]=pcr(x,c,numc,yun);

函数 pcr 为间接校正的主成分回归方法

【例子】

[p,cun]=pcr(x,c,numc,yun);

输入参数:

x 为混合物量测波谱矩阵;

c 为相应的混合物浓度矩阵;

numc 为应取的主成分数;

yun 为未知混合物量测波谱矩阵或矢量。

输出参数:

p 为计算所得的回归系数矩阵或矢量;

cun 为计算所得的未知混合物量测波谱矩阵或矢量所对应的浓度矩阵或矢量。

```
% Check that matrix (X) and left hand side (y) have compatible dimensions
[n,p]=size(x);
[n1,collhs]=size(c);
if n ~=n1,
   error('The number of rows in Y must equal the number of rows in X.');
end

[s,v,d]=svd(x);
p=c * d(:,1:numc) * inv(v(1:numc,1:numc)) * s(:,1:numc)';
cun=p * yun;
```

29. function [p,q,w,b,t,u,x,y]=pls(a,c,maxrank)

函数 PLS 为偏最小二乘子程序

【例子】

(1) [P,Q,W,B]=PLS(A,C,MAXRANK)

输入参数:

A 为用于训练集量测矩阵;

C 为用于训练集浓度矩阵或其他响应矩阵;

MAXRANK 为可选用的最大主成分数。

输出参数:

P 为量测矩阵的载荷因子矩阵;

Q 为浓度或其他响应矩阵的载荷因子矩阵;

W 为权重矩阵;

B 为内部相关矢量。

(2) [P,Q,W,B,T,U,X,Y]=PLS(A,C,MAXRANK)

输入参数：

A 为用于训练集量测矩阵；

C 为用于训练集浓度矩阵或其他响应矩阵；

MAXRANK 为可选用的最大主成分数。

输出参数：

P 为量测矩阵的载荷因子矩阵；

Q 为浓度或其他响应矩阵的载荷因子矩阵；

W 为权重矩阵；

B 为内部相关矢量；

T 为量测矩阵的得分因子矩阵；

U 为浓度或其他响应矩阵的得分因子矩阵；

X 为量测矩阵的残差矩阵；

Y 为浓度或其他响应矩阵的残差矩阵。

【程序】

```
[m,n]=size(a);
[m,l]=size(c);
if nargin==3,
i=min([n,m,maxrank]);
else i=min([n m]);
end

% x=a;
% y=c;

for h=1:i,
   xy=x' * y;
   [pt,s,qt]=svd(xy' * xy,0);
   q(h,:)=qt(:,1)';
   w(:,h)=(xy * qt(:,1))/sqrt(s(1,1));
   t(:,h)=x * w(:,h);
   tsqr=t(:,h)' * t(:,h);

   u(:,h)=y * qt(:,1);
   p(h,:)=t(:,h)' * x / tsqr;
   b(1,h)=(u(:,h)' * t(:,h)) / tsqr;

   x=x - (t(:,h) * p(h,:));
   y=y - b(h) * t(:,h) * q(h,:);
end
```

30. function [press,minlv]=Tcvpls(X,Y,maxnumb,selectnumb);

函数 Tcvpls 用于交叉效验(cross-validation)计算，本程序是调用的 PLS 子程序，同理，调用 PCR 也是可行的。

【例子】

[press,minlv]=Tcvpls(X,Y,maxnumb,selectnumb);

输入参数

X 为自变量矩阵(independent data matrix),在间接校正中即为混合物量测或校正矩阵;

Y 为因变量矩阵(dependent data matrix),在间接校正中即为混合物浓度矩阵;

maxnumb 为用户给定的最大主成分数;

selectnumb 为用户给定的每次训练时实际要做到的主成分数;

输出参数

press 给定主成分数时的模型预测总误差(prediction error of squared sum);

minlv 模型预测总误差最小点,用以决定最佳主成分数。

【程序】

```
[m,n]=size(X);maxnumb=rank(X);AA=selectnumb;
HHH=[];TTT=[];PYR=[];

  for k=1:m
      if k==1
      X1=X([2:m],:);
      Y1=Y([2:m],:);

      elseif k==m
         X1=X([1:m-1],:);
           Y1=Y([1:m-1],:);
      else X1=X([1:k-1,k+1:m],:);
              Y1=Y([1:k-1,k+1:m],:);
    end

  for i=1:n
     for j=1:m-1
         X1Z(j,i)=X1(j,i)-mean(X1(:,i));
     end
  end

  Y1Z=Y1-mean(Y1);
  XX=X1Z;
  YR=Y1Z;
  W=[];T=[];C=[];U=[];P=[];Q=[];H=[];R=[];
  for i=1:AA
    w=XX'*YR/sqrt(YR'*XX*XX'*YR);
    t=XX*w;
    c=YR'*t/(t'*t);
    u=YR*c/(c'*c);
    p=XX'*t/(t'*t);
    q=YR'*u/(u'*u);
```

```
    b=u' * t/(t' * t);
  XX=XX-t * p';YR=YR-b * t * c';
    W(:,i)=w;
    T(:,i)=t;
    C(:,i)=c;
    U(:,i)=u;
    P(:,i)=p;
    Q(:,i)=q;
  end
    H(:,1)=W(:,1);R=eye(n)-W(:,1) * P(:,1)';
  for j=2:AA
        H(:,j)=R * W(:,j);
        R=R * (eye(n)-W(:,j) * P(:,j)');
  end

  HHH=[HHH H];
  TTT=[TTT T];
end

for k=1:m

    a=(k-1) * AA;
    pyr=[];

    for h=1:AA

Bp=HHH(:,a+1:a+h) * inv(TTT(:,a+1:a+h)' * TTT(:,a+1:a+h)) * HHH(:,a+1:a+b)' * X1Z' *
Y1Z;
        CC=(mean(Y1)-mean(X1) * Bp);
        Yrp=CC+X(k,:) * Bp;
        pyr=[pyr Yrp];
    end
        PYR=[PYR;pyr];
    end
    pressr=[];
            press=[];
    for i=1:AA
        cvr=(Y-PYR(:,i))' * (Y-PYR(:,i));
        press=[press,cvr];
    end
        plot(press,'g')
    [pressr,minlv]=min(press)
```

31. function [x,w]=nnls(E,f,tol)

函数 NNLS 用于解决非负最小二乘问题，即所得回归估计矢量为非负矢量。

【例子】

X=NNLS(A,b)

输入参数:

A 为一个(n×p)阶校正矩阵;

b 为一(n×1)阶相应矢量。

输出参数:X 为计算所得的加权最小二乘估计矢量或矩阵(p×d)。

参见 LSCOV,SLASH.

参考文献:

Lawson and Hanson,"Solving Least Squares Problems",Prentice-Hall,1974

【程序】

```
if nargin<2,error('Not enough input arguments.');end
if ~isreal(E) | ~isreal(f),error('A and b must be real.');end

% initialize variables
if nargin < 3
    tol=10 * eps * norm(E,1) * length(E);
end
[m,n]=size(E);
P=zeros(1,n);
Z=1:n;
x=P';
ZZ=Z;
w=E' * (f-E * x);

% set up iteration criterion
iter=0;
itmax=3 * n;

% outer loop to put variables into set to hold positive coefficients
while any(Z) & any(w(ZZ) > tol)
    [wt,t]=max(w(ZZ));
    t=ZZ(t);
    P(1,t)=t;
    Z(t)=0;
    PP=find(P);
    ZZ=find(Z);
    nzz=size(ZZ);
    EP(1:m,PP)=E(:,PP);
    EP(:,ZZ)=zeros(m,nzz(2));
    z=pinv(EP) * f;
    z(ZZ)=zeros(nzz(2),nzz(1));
% inner loop to remove elements from the positive set which no longer belong
    while any((z(PP) <=tol))
        iter=iter + 1;
```

```
        if iter > itmax
            error(['Iteration count is exceeded.',...
                  'Try raising the tolerance.'])
        end
        QQ=find((z <=tol) & P');
        alpha=min(x(QQ)./(x(QQ) - z(QQ)));
        x=x + alpha * (z - x);
        ij=find(abs(x) < tol & P' ~= 0);
        Z(ij)=ij';
        P(ij)=zeros(1,length(ij));
        PP=find(P);
        ZZ=find(Z);
        nzz=size(ZZ);
        EP(1:m,PP)=E(:,PP);
        EP(:,ZZ)=zeros(m,nzz(2));
        z=pinv(EP) * f;
        z(ZZ)=zeros(nzz(2),nzz(1));
    end
    x=z;
    w=E' * (f - E * x);
end
```

32. function sida=m _ esti(x,y,iterNum);

函数 m _esti 为稳健的 M _估计程序。

【例子】

sida=m _ esti(x,y,iterNum);

输入参数：

x 为一个(n×p)阶校正矩阵；

y 为一(n×1)阶相应矢量；

iterNum 为迭代次数。

输出参数：sida 为计算所得的稳健估计矢量。

【程序】

```
[n,m]=size(x);
sida1=inv(x' * x) * x' * y;
sida1
residu=y-x * sida1;
plot(1:n,x * sida1,'*');
pause
t=1:n;
plot(t,residu,'+');
residu
median(abs(residu))
    for j=1:n
  if abs(residu(j))>=1.5 * median(abs(residu))
      residu(j)=1.5 * median(abs(residu)) * sign(residu(j));
```

```
else
residu(j)=residu(j);
end
  end
  residu;
%  for j=1:n
%  if abs(residu(j)/(1.5 * median(abs(residu))))<=3.1416
%        weigts(j)=sin(residu(j))/residu(j);
%  else
%  weigts(j)=0;
%  end
%    end
sida2=inv(x' * x) * x' * residu;
sida=sida1+sida2;
for k=1:iterNum
   resi=y-x * sida;
  resi;
  t=1:n;
  plot(t,resi,'+');
  pause
  for j=1:n
  if abs(resi(j)/median(abs(resi)))<=3.1416
       weigts(j)=sin(resi(j))/resi(j);
  else
  weigts(j)=0;
  end
   end
  weigts;
  w=eye(n);
  for i=1:n
  w(i,i)=weigts(i);
  end
  sida=inv(x' * w * x) * x' * w * y
  plot(t,x * sida,' * ',t,y,'o',t,y,'          ');
pause
end
end
```

33. function [x,y]=simu1d(numc,h1,cen1,sgm1,h2,cen2,sgm2,noise,c);

函数 simu1d 可用于产生白色分析体系的模拟校正矩阵 x 和模拟混合物响应矢量 y。每一个组分光谱均由两个高斯峰所组成，而模拟混合物响应矢量 y 则由给定组分数(numc)的纯光谱由给定的浓度(c)，通过线性加合而成。

【例子】

(1) [x,y]=simu1d(numc,h1,cen1,sgm1,h2,cen2,sgm2,noise,c);

输入参数：

numc 为体系的组分数；

h1为组成纯组分光谱的第一个高斯峰的峰高矢量(行矢量)；

cen1为组成纯组分光谱的第一个高斯峰的峰位置矢量(行矢量)；

sgm1为组成纯组分光谱的第一个高斯峰的标准偏差矢量(行矢量)；

h2为组成纯组分光谱的第二个高斯峰的峰高矢量(行矢量)；

cen1为组成纯组分光谱的第二个高斯峰的峰位置矢量(行矢量)；

sgm1为组成纯组分光谱的第二个高斯峰的标准偏差矢量(行矢量)；

noise 为加入模拟混合物响应矢量 y 的服从正态分布的误差矢量的标准差；

c 为包括各组分的相对浓度的矢量(列向量)。

输出参数：

x 为包含所得的标准纯光谱的(80×numc)阶的校正矩阵；

y 为(80×1)阶的模拟混合物响应矢量。

(2) 使用示例

```
[x1,y1]=simuld(3,[1 1 1],[25 25 35],[7 7 8],[.5 .6 .7],[35 36 50],[8 7.6 6],.04,[.5 .7 .6]');
```

【程序】

```
for i=1:numc
  for j=1:80
      x(j,i)=h1(i)*exp(-(j-cen1(i))^2/(2*sgm1(i)^2))-h2(i)*exp(-(j-cen2(i))^2/(2*sgm2
(i)^2));
  end
end

y=x*c;
y=y+randn([80 1])*noise;
```

34. function [eig]=rafa(x,y,alf_range,stepwise);

函数 rafa 为一简易的广义秩消失因子分析法，其结果为一对数特征值对相对浓度系数的变化图，其对数特征值的极小点所对应的相对浓度系数值即为所求。

【例子】

[eig]=rafa(x,y,alf_range,stepwise);

输入参数：

x 为混合物的二维量测矩阵；

y 为标准物的二维量测矩阵；

alf_range 为给定相对浓度系数需计算的区间矢量；

stepwise 为计算步长。

输出参数：eig 为计算所得特征值矩阵。

【程序】

```
seri=[alf_range(1):stepwise:alf_range(2)];
iternum=ceil((alf_range(2)- alf_range(1))/stepwise);
[n,m]=size(seri)
if m>iternum
    iternum=iternum+1
end
pause
eig=[];
```

```
for i=1:iternum
    Re=x-(alf_range(1)+i*stepwise)*y;
    [s,v,d]=svd(Re);
    eig(:,i)=diag(v(1:10,1:10));
end
plot(seri,log10(eig)');
```

35. function [kons]=lorbm(X,S,C,rank);

函数 lorbm 为一广义秩消失因子分析法的 Lorber 方法，其结果为混合物中所含标准物质相对浓度系数值。

【例子】

[kons]=lorbm(X,S,C,rank);

输入参数:

X 为混合物的二维量测矩阵；

S 为标准物纯光谱矢量；

C 为标准物纯色谱矢量；

rank 为混合物的二维量测矩阵的化学秩，及组分数。

输出参数:kons 为计算所得相对浓度系数。

【程序】

```
function [kons]=lorbm(X,S,C,rank);

[U,Seig,V]=svd(X);
Ured=U(:,1:rank);
Sred=Seig(1:rank,1:rank);
Vred=V(:,1:rank);
D=Ured*Sred*Vred';
a=Ured'*C;
b=Vred'*S;
M=a*b'*inv(Sred);
[eigvekt,eigval]=eig(M)
temp=sort(diag(eigval));
eigval(1,1)=1/eigval(1,1);
Kons=1/temp(rank,1)
```

36. function [con]=rbl(x,y,ranky,num,noise)

函数 rbl 为残差双线性分解算法子程序。

【例子】

[con]=rbl(x,y,ranky,num,noise)

输入参数:

x 为含待测的纯组分的三阶矩阵，即 x(i,j,k)表示第 k 待测组分量测矩阵中第 i,j 位置上的量测值；

y 为混合物量测矩阵；

ranky 为混合物量测矩阵的秩，或混合物体系的组分数；

num 待测纯组分的数目；

noise 为量测误差，用作程序终止条件。

输出参数:con 为待测组分的相对浓度；

```
[rx,cx,bx]=size(x);
[ry,cy]=size(y);

if cx ~=cy
   error('Requires the inputs to have the same number of columns.');
end

if rx ~=ry
   error('Requires the inputs to have the same number of rows.');
end

y_re=reshape(y,ry*cy,1);
x_re=reshape(x,rx*cx,bx);

b=lsq(x_re,y_re);
%pause

%calculation of the initial residual matrix
[k,k1]=size(b);
res0=y;
for i=1:k-1
   res0=res0-b(i)*x(:,:,i);
end
[s1,v1,d1]=svd(res0);

%calculation of the residual tolerance
res_tol=res0-s1(:,1:(ranky-num))*v1(1:(ranky-num),1:(ranky-num))...
    *d1(:,1:(ranky-num))';
tol=norm(res_tol)/15
tol1=0;
%pause
tt=tol-tol1;
%comparison tol with the noise level
whilett>=.00001 & tol>noise
   res0=s1(:,1:(ranky-num))*v1(1:(ranky-num),1:(ranky-num))...
    *d1(:,1:(ranky-num))';
   resi=y-res0;
   resi_re=reshape(resi,ry*cy,1);

   x_re=reshape(x,rx*cx,bx);
   b=lsq(x_re,resi_re);
   [k,k1]=size(b);
   resi=y;
```

```
   for j=1:k-1
       resi=resi- b(j) * x(:,:,j);
   end

   [s1,v1,d1]=svd(resi);
   res_tol=resi-s1(:,1:(ranky-num)) * v1(1:(ranky-num),1:(ranky-num))...
       * d1(:,1:(ranky-num))';
   tol1=tol;
   tol=norm(res_tol)/15;
   tt=tol1-tol;
%  res0=s1(:,1:(ranky-num)) * v1(1:(ranky-num),1:(ranky-num))...
%   * d1(:,1:(ranky-num))';
   pause
end

con=b(1:num,1)
display('program is finished')
```

37. function [bak1,bak2,bk1,bk2]=bakremo(action,x);

函数 bakremo 可用于二维数据的背景扣除，该程序应与函数 name 联合使用。该程序提供了一个图界面，用户就直接用鼠标在图上将峰簇前一段零组分区与峰簇后一段零组分区点出即可。

【例子】

[bak1,bak2,bk1,bk2]=bakremo(action,x);

输入参数：

action 为一字符串参数，一般以'start'代入即可；

x 为1个二维数据矩阵，该矩阵的每一行为一个光谱；每一列为一个色谱(或一滴定曲线，或一动力学系统的连续时间记录，等)；

输出参数：

bak1为用主成分背景扣除方法而得的扣除了背景的数据矩阵(参见文献;)在程序运行过程中，按下'BakSub'即可。

bak2为用经典一维背景扣除方法而得的扣除了背景的数据矩阵，在程序运行过程中，按下'ClaBC'即可。

bk1为峰簇前一段零组分区的背景光谱矢量；

bk2为峰簇后一段零组分区的背景光谱矢量。

【程序】

```
% possible actions:
%  'start'
%     'clabc'
%     'ReDo'
%  'remove'
%  'quit'

load dataname;
global BACKCORR_DAT;
global bak1;
```

```
global bak2;
global bk1;
global bk2;

if strcmp(action,'start')
    % graphics initialization

    clf reset.
    set(gcf,'Units','normalized','backingstore','off');

    % Create initial plot
    [n,m]= size(x);
    max _ nums=n;
    max _ numc=m;
    min _ nums=1;
    min _ numc=1;
        ymax=max(x);
    Wave=10;
    Time=10;

    bak1=zeros(n,m);
    bak2=zeros(n,m);
    clabcstr=str2mat(['[bak1,bak2,bk1,bk2]=bakremo1("ClaBC", 's ');']);
        clabc _ button=uicontrol('Style','Pushbutton','Position',[.87 .26 .12 .07],…
        'Units','normalized','Callback',clabcstr,'String','ClaBC');
        removestr=str2mat(['[bak1,bak2,bk1,bk2]=bakremo1("remove",' s ');']);
    remove _ button=uicontrol('Style','Pushbutton','Position',[.87 .18 .12 .07],…
        'Units','normalized','Callback',removestr,'String','BakSub');
        redostr=str2mat(['[bak1,bak2,bk1,bk2]=bakremo1("start",' s ');']);
    repeat _ button=uicontrol('Style','Pushbutton','Position',[.87 .1 .12 .07],…
        'Units','normalized','Callback',redostr,'String','ReDo');
    quitstr=str2mat(['bakremo1("quit")']);
    close _ button=uicontrol('Style','Pushbutton','Position',[.87 .02 .12 .07],…
        'Units','normalized','Callback',quitstr,'String','Quit');

        ax _ chro=axes('Position',[.10 .75 .7 .20]);
        Chro _ line=plot(1:max _ nums,mean(x'),'r','EraseMode','xor');
        %axis([-8 8 -.5 2]);
        grid;
        %title('use mouse to select the desired zero-component regions to compare');
        ylabel('A');
        xlabel('Retention Time');

        %the first part
```

```
    [xx,yy]=ginput(2)
low=floor(xx(1));
if low < 1
  low=1;
end
high=ceil(xx(2));
if high > n
   high=n;
end

   [u,s,v]=svd(x(low:high,:));
figure(1);
[xx,yy]=bar(low:high,x(low:high,Wave));
hold on; plot(xx,yy,'m');
   bk1=v(:,1) * s(1,1) * u(1,1);
   ax_pca1=axes('Position',[.10 .25 .30 .35]);
   pca_line1=plot(1:max_numc,v(:,1:2),'EraseMode','xor');
hold on;
   plot(1:max_numc,v(:,1),'+');

   ylabel('Loadings');
   xlabel('Wavelength');

   uicontrol('style','text','pos',[.83 .92 .16 .05],...
     'Units','normalized','BackgroundColor','black',...
     'ForegroundColor','white','String','EigVals 1:');

   uicontrol('style','text','pos',[.83 .87 .16 .05 ],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String',num2str(s(1,1)));

   uicontrol('style','text','pos',[.83 .82 .16 .05 ],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String',num2str(s(2,2)));

    uicontrol('style','text','pos',[.18 .08 .16 .05],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String',[' Rel:',num2str(low),'--',
num2str(high)]);

%the second part

 ax_chro=axes('Position',[.10 .75 .7 .20]);
```

```
Chro _ line=plot(1:max _ nums,mean(x'),'y','EraseMode','xor');

%axis([-8 8 -.5 2]);
  grid;
  %title('use mouse to select the desired zero-component regions to compare');
  ylabel('A');
  xlabel('Retention Time');
  [xx1,yy1]=ginput(2)

low1=floor(xx1(1));
if low1 < 1
  low1=1;
end
high1=ceil(xx1(2));
if high1 > n
  high1=n;
end

[u,s,v]=svd(x(low1:high1,:));
bk2=v(:,1) * s(1,1) * u(1,1);
figure(1);
 [xx1,yy1]=bar(low1:high1,x(low1:high1,Wave));
 hold on; plot(xx1,yy1,'m');
 ax _ pca2=axes('Position',[.55 .25 .30 .35],'XLim',[0
max _ nums],'YLim',[-.5 2]);
 pca _ line2=plot(1:max _ numc,v(:,1:2),'EraseMode','xor');
 hold on;
   plot(1:max _ numc,v(:,1),'+');

 ylabel('Loadings');
 xlabel('Wavelength');

 uicontrol('style','text','pos',[.83 .70 .16 .05],...
     'Units','normalized','BackgroundColor','black',...
     'ForegroundColor','white','String','EigVals 2:');

 uicontrol('style','text','pos',[.83 .65 .16 .05 ],...
     'Units','normalized','BackgroundColor','black',...
     'ForegroundColor','white','String',num2str(s(1,1)));

 uicontrol('style','text','pos',[.83 .60 .16 .05 ],...
     'Units','normalized','BackgroundColor','black',...
     'ForegroundColor','white','String',num2str(s(2,2)));
```

```
uicontrol('style','text','pos',[.58 .08 .20 .05],...
     'Units','normalized','BackgroundColor','black',...
     'ForegroundColor','white','String',[' Re2:',num2str(low1),'   ',num2str(high1)]);

b1=bk1-min(bk1);
b2=bk2-min(bk2);
cong=b1' * b2/norm(b1)/norm(b2);

uicontrol('style','text','pos',[.25 .02 .50 .05],...
     'Units','normalized','BackgroundColor','black',...
     'ForegroundColor','white','String',['Congruence coefficient:',
num2str(cong)]);
drawnow;

BACKCORR_DAT = [Wave; Time; min_nums; max_nums;min_numc; max_numc; 0; 0; ...
low;high;low1;high1];

elseif strcmp(action,'remove')

low=BACKCORR_DAT(9);
high=BACKCORR_DAT(10);
low1=BACKCORR_DAT(11);
high1=BACKCORR_DAT(12);
max_nums=BACKCORR_DAT(4);
max_numc=BACKCORR_DAT(6);
min_nums=BACKCORR_DAT(3);
min_numc=BACKCORR_DAT(5);
Wave=BACKCORR_DAT(1);
Time=BACKCORR_DAT(2);

for j=low:high
   tia(j-low+1)=x(j,Wave);
end

for j=low1:high1
   tih(j-low1+1)=x(j,Wave);
end

tiab=[tia tib];
rt=[low:high low1:high1];

bb=lsq(rt',tiab');
b_est=bb(2)+bb(1) * (1:max_nums);
b_est=b_est';
```

```
      com1=bk1(Wave);
      com2=b_est(low);

  comd=com1-com2;

  for i=1:max_numc
        for j=1:max_nums
     bak1(j,i) = x(j,i)-b_est(j)-bk1(i)+b_est(low)+comd;
        end
  end

elseif strcmp(action,'ClaBC')

  low=BACKCORR_DAT(9);
  high=BACKCORR_DAT(10);
  low1=BACKCORR_DAT(11);
  high1=BACKCORR_DAT(12);
  max_nums=BACKCORR_DAT(4);
  max_numc=BACKCORR_DAT(6);
  min_nums=BACKCORR_DAT(3);
  min_numc=BACKCORR_DAT(5);
  Wave=BACKCORR_DAT(1);
  Time=BACKCORR_DAT(2);

  for i=1:max_numc

     for j=low:high
     tia(j-low+1)=x(j,i);
  end

  for j=low1:high1
     tib(j-low1+1)=x(j,i);
  end

  tiab=[tia tib];
  rt=[low:high low1:high1];

  bb=lsq(rt',tiab');

  b_est=bb(2)+bb(1)*(1:max_nums);
  b_est1=b_est';
  bak2(:,i) = x(:,i)-b_est1;
```

```
  end
elseif strcmp(action,'quit')
  clear global BACKCORR_DAT;
  delete(gcf);
  delete dataname.mat;
end
```

38. function [puprof,puspec]=als(x,pusini,iternum);

函数 als 为交换最小二乘(alterative least squares)子程序。该程序在迭代过程中提供了一个图界面，用户就直接用鼠标在图上将光谱或色谱中为负的部分点出即可。

【例子】

[puprof,puspec]=als(x,pusini);

输入参数.

x 为量测二维数据矩阵，其每一行为一个光谱，每一列为一个色谱；

pusini 为初始迭代光谱矩阵；

iternum 迭代次数。

输出参数：

puprof 计算所得的纯色谱矩阵；

puspec 计算所得的纯光谱矩阵。

【程序】

```
[m,n]=size(pusini);
puprof=inv(pusini' * pusini) * pusini' * x;
puspec=pusini;

for i=1:iternum

puprof=inv(puspec' * puspec) * puspec' * x;
for j=1:n
plot(puprof(j,:)');
disp('Use the mouse pointer to mark the region which should be zeros')
   disp('Mark the lower index first')
   [p1,p2]=ginput(2);
   low=floor(p1(1));
   if low < 1
     low=1;
   end
   high=floor(p1(2));
   if high > m
     high=m;
   end
   if high>low
   puprof(j,low:high)=zeros([1 (high-low+1)]);
   else
   end
 end
```

```
plot(puprof')
pause
clg

puspec=x * puprof' * inv(puprof * puprof');
for j=1:n
plot(puspec(:,j));
disp('Use the mouse pointer to mark the region which should be zeros')
   disp('Mark the lower index first')
   [p1,p2]=ginput(2);
   low=floor(p1(1));
   if low < 1
     low=1;
   end
   high=ceil(p1(2));
   if high > m
     high=m;
   end
 puspec(low:high,j)=zeros([(high-low+1) 1 ]);
 end
 plot(puspec)
 pause
 clg
 end
```

39. function [eig,eig1]=evol(x,tim);

函数 evol 为渐进因子分析子程序。它将给出计算所得的前10个特征值的对保留时间对数变化图，以确定各组分的流出和消失点。

【例子】

[eig]=evol(x,tim);

输入参数：

x 为量测二维数据矩阵，其每一行为一个光谱，每一列为一个色谱；

tim 为保留时间矢量或保留时间方向的点数矢量。

输出参数：

eig 前向计算所得的每一时间点所对应的前10个特征值所构成矩阵；

eig1 反向计算所得的每一时间点所对应的前10个特征值所构成矩阵。

【程序】

```
[m,n]=size(x);
eigt=zeros([m m]);

%calculation along with the forward direction
for i=2:m
[s,v,d]=svd(x(1:i,:));
tem=[diag(v);zeros([m-i,1])];
```

```
eigt(i,:)=eigt(i,:)+tem';
end
eig=eigt(:,1:10);
plot(tim,log10(eig(:,1:10)))
xlabel('Retention time')
ylabel('log10(Eigvalues)')
hold on

%calculation along with the reverse direction
eigt=zeros([m m]);
for i=m-1:-1:1
[s,v,d]=svd(x(i:m,:));
tem=[diag(v);zeros([i-1,1])];
eigt(i,:)=eigt(i,:)+tem';
end
plot(tim,log10(eigt(:,1:10)))
eig1=eigt(:,1:10);
xlabel('Retention time')
ylabel('log10(Eigvalues)')
```

40. function [SS]=spc(X);

函数 spc 为子窗口因子分析方法子程序。其方法的基本思路和算法可参见第七章。

本程序提供了一个友好界面，用户只需用鼠标定义两个相邻多组分区域，且该两个多组分区域都含有一个共同组分，则可得到该组分的纯光谱。

【例子】

[SS]=spc(X);

输入参数：X 为二维数据矩阵

输出参数：SS 为所得纯光谱。

```
color=['ymcrgbw'];
slutt=1;
[n,m]=size(X);
mx=mean(X');
k=1;
SS=[];
if nargin==1
    winsize=input('Please input the size of moving window  ');

    j=1;
    for i=1:n-winsize+1
    s=svd(X(i:i+winsize-1,:));
    eigs(j,:)=s';
    j=j+1;
    end
    eigs=log10(eigs);
```

```
    end
    subplot(212),plot(eigs)
    title('Rank map')
    ylabel('Log(EV)');
    xlabel('Retention time');
    subplot(211),plot(mx);

while slutt == 1

    disp('Use the mouse pointer to mark the suspected regions')

    [x,y]=ginput(4);
    for i=1:4
        if x(i)<1
            x(i)=1;
        elseif x(i)>n
            x(i)=n;
        else
            x(i)=ceil(x(i));
        end
    end

    i=x(1):x(2);
    j=x(3):x(4);
    subplot(211),hold on;
    subplot(211),bar(i,mx(i),color(1));
    subplot(211),bar(j,mx(j),color(2));
    pause
    [E,s,v]=svd(X(i,:)');[F,s,v]=svd(X(j,:)');
    clear s v;
    E=E(:,1:3);F=F(:,1:3);
    [V,D]=eig(E' * F * F' * E);
    [v,d]=eig(F' * E * E' * F);
    % decide which eigvector to use
    if d(2,2)>d(1,1)
        s=F * v(:,2);
    else
        s=F * v(:,1);
    end
    if D(2,2)>D(1,1)
        S=E * V(:,2);
    else
        S=E * V(:,1);
    end
```

```
    s=[s S];
    for i=1:2
        if max(s(:,i))<abs(min(s(:,i)))
            s(:,i)=-s(:,i);
        end
    end
    figure(2)
    subplot(211)
    plot(s);
    title(num2str(D(1,1)));
    xlabel(num2str(D(2,2)));
    subplot(212)
    plot(s(:,1)--s(:,2));grid
    clear E F V v d S
    answer=input('Is this region OK? (1=Yes, 0=No)  ');

    if answer == 1
        selind(k,:)=[x(1) x(2) x(3) x(4)];
        k=k+1;
        SS=[SS s(:,1)];
    end
    delete(figure(2))
    slutt=input('Do you want to continue testing? (1=Yes, 0=No)  ');
end
```

41. function [pure]=Orthopro(x);

函数 Orthopro 为正交投影分辨子程序。其方法的基本思路和算法可参见第六章。

本程序提供了一个友好界面，用户只需用鼠标定义两个相邻多组分区域，且该两个多组分区域都含有一个共同组分，则可得到该组分的纯光谱。

【例子】

[pure]=Orthopro(x);

输入参数：x 为二维数据矩阵。

输出参数：pure 为计算所得的纯色谱矩阵。

【程序】

```
Numtotal=input('Please input the number of components in the system:  ');
Num=Numtotal-1;
[n,m]=size(x);
max_nums=n;
max_numc=m;
min_nums=1;
min_numc=1;

winsize=input('Please input the window size of rank analysis:  ');
j=winsize;
  for i=1:max_nums-winsize+1
```

```
    [u,s,v]=svd(x(i:i+winsize-1,:));
    cons(j,:)=(diag(s))';
    j=j+1;
    end

    cons=log10(cons);
kk=0
slutt=1;
    while(slutt==1 & kk<Numtotal)

    figure(1)
    p1=plot(1:max_nums,cons);
    grid;
    title('use mouse to select the desired regions to resolve');
    ylabel('Log(Eigs)');
    xlabel('Retention Time');
    [xx,yy]=ginput(2)
    low=floor(xx(1));
    if low < 1
      low=1;
    end
    high=ceil(xx(2));
    if high > n
      high=n;
    end

    slutt1=input('Do you want to try to mark another region? (0=No,1=Yes) ');
    if slutt1==1
      figure(1)
     p1=plot(1:max_nums,cons);
     grid;
     title('use mouse to select the desired regions to resolve');
     ylabel('Log(Eigs)');
     xlabel('Retention Time');

     [xx,yy]=ginput(2)
     low1=floor(xx(1));
     if low1 < 1
       low1=1;
     end
     high1=ceil(xx(2));
     if high1 > n
       high1=n;
     end
```

```
    xab=[x(low:high,:);x(low1:high1,:)];
    [u,s,v]=svd(xab);
    re=zeros([max_nums,max_numc]);
    re=x-x*v(:,1:Num)*v(:,1:Num)';
    else
    [u,s,v]=svd(x(low:high,:));
    re=zeros([max_nums,max_numc]);
    re=x-x*v(:,1:Num)*v(:,1:Num)';
end

kk=kk+1;

    for k=1:max_nums
    pure(k,kk)=sum(re(k,:));
    end

if sum(pure(:,kk))<0
    pure(:,kk)=-pure(:,kk);
end

pure(:,kk)=pure(:,kk)./norm(pure(:,kk));
    figure(2)
    p2=plot(1:max_nums,pure(:,kk));
    hold on

    slutt=input('Do you want to try to resolve another component? (0=No,1=Yes) ');
end
delete(p1)
hold off
```

42. function b=lsq(x,y)

函数 lsq 为简单最小二乘计算程序，主要用于其他函数调用。

【例子】

b=lsq(x,y)

输入参数：

x 为一(m×n)阶数据矩阵；

y 为一(m×1)阶列矢量。

输出参数：b 为一(n×1)阶的列矢量，即计算所得的回归系数。

【程序】

```
[m,n]=size(x);
% b=zeros(n+1);
aa=mean(x);

ba=mean(y);
```

```
for i=1:m
x(i,:)=x(i,:)-aa;
y(i)=y(i)-ba;
end
b=inv(x'*x)*x'*y;
s=ba-aa*b;
b=[b;s];
end
```

43. function [x]=movpca(action,x,in1)

函数 movpca 可用于动态局部主成分分析，通过滑动条即可对数据矩阵的任一部分进行固定窗口的主成分分析。

【例子】

[x]=movpca(action,x,in1)

输入参数：

action 为一字符串参数，一般以'start'代入即可。

x 为1个二维数据矩阵，该矩阵的每一行为一个光谱；每一列为一个色谱（或一滴定曲线，或一动力学系统的连续时间记录，等）。

【程序】

```
% possible actions:
% 'start'
%   'ChanWav'  - in1=1 ==> from slider, in1=2 ==> from edit text
%   'ChanRet'  - in1=1 ==> from slider
% 'redraw'     - in1=1 ==> from Wave, in1=2 ==> from Time
% 'done'

slutt=1;
%[n,m]=size(x);

global BACKCORR_DAT

if strcmp(action,'start'),
% graphics initialization
Contr=input('Please input the size of moving window: ');

clf reset;
set(gcf,'Units','normalized','backingstore','off');

% Create initial plot
[n,m]=size(x);
max_nums=n;
max_numc=m;
min_nums=1;
min_numc=1;
```

```
ymax=max(x);
Wave=10;
Time=10;

    Wave_text=uicontrol('Style','text','Position',[.18 .02 .38 .07],...
        'Units','normalized','BackgroundColor','black',...
        'ForegroundColor','white','String','Wavelength:');

  uicontrol('style','text','pos',[.14 .07 .02 .05],...
        'Units','normalized','BackgroundColor','black',...
        'ForegroundColor','white','String',num2str(min_numc));

  uicontrol('style','text','pos',[.74 .07 .06 .05 ],...
        'Units','normalized','BackgroundColor','black',...
        'ForegroundColor','white','String',num2str(max_numc));

  Wave_field =uicontrol('Style','edit','Position',[.59 .02 .12 .07],...
        'Units','normalized','String',num2str(Wave),'Userdata',x,...
        'CallBack','movpca("ChanWav",x,2);movpca("redraw",x,1);');

  Wave_slider=uicontrol('Style','slider','Position',[.15 .1 .6 .04]'...
        'Units','normalized','Value',Wave,'Max',max_numc,'Min',min_nums,'UserData',x,...
        'Callback','movpca("ChanWav",x,1);movpca("redraw",x,1);');

  uicontrol('style','text','pos',[.14 .65 .02 .05],...
        'Units','normalized','BackgroundColor','black',...
        'ForegroundColor','white','String',num2str(min_nums));

  uicontrol('style','text','pos',[.90 .65 .06 .05 ],...
        'Units','normalized','BackgroundColor','black',...
        'ForegroundColor','white','String',num2str(max_nums));

  Ret_slider=uicontrol('Style','slider','Position',[.15 .7 .75 .04],...
        'Units','normalized','Value',Time,'Max',max_nums,'Min',min_nums,'UserData',x,...
        'Callback','movpca("ChanRet",x);movpca("redraw",x,2);');

  close_button=uicontrol('Style','Pushbutton','Position',[.85 .02 .12 .07],...
        'Units','normalized','Callback','movpca("done")','String','Done');

%subplot(222),plot(x(:,10))

  %frequency domain
  ax_chro=axes('Position',[.16 .75 .7 .20]);
  Chro_line=plot(1:max_nums,x(:,Wave),'EraseMode','xor');
```

```
%axis([--8 8 -.5 2]);
grid;
ylabel('A');
xlabel('Retention Time');
%ax_pca(i)=zeros(Contr);
  %Chro_line(i)=zeros(Contr);
  [u,s,v]=svd(x(10:10+Contr-1,:));
  %for i=1:Contr
  ax_pca=axes('Position',[.16 .25 .5 .35]);
  pca_line=plot(1:max_numc,v(:,1:Contr),'EraseMode','xor');
  %end
  ylabel('A');
  xlabel('Retention Time');
  if Contr==2,
  str1=['EV1: ',num2str(s(1,1))];
  str2=['EV2: ',num2str(s(2,2))];
  str3=['          '];
  elseif Contr>=3,
  str1=['EV1: ',num2str(s(1,1))];
  str2=['EV2: ',num2str(s(2,2))];
  str3=['EV3: ',num2str(s(3,3))];
  end
  str4=['Region: ',num2str(Time),'-',num2str(Time+Contr-1)];
  uicontrol('style','text','pos',[.70 .35 .28 .05],...
       'Units','normalized','BackgroundColor','black',...
       'ForegroundColor','white','String',str1);
  uicontrol('style','text','pos',[.70 .28 .28 .05],...
       'Units','normalized','BackgroundColor','black',...
       'ForegroundColor','white','String',str2);

  uicontrol('style','text','pos',[.70 .22 .28 .05],...
       'Units','normalized','BackgroundColor','black',...
       'ForegroundColor','white','String',str3);
  uicontrol('style','text','pos',[.70 .42 .28 .05],...
       'Units','normalized','BackgroundColor','black',...
       'ForegroundColor','white','String',str4);
  %for i=1:Contr
  set(Chro_line,'ButtonDownFcn','movpca(''redraw'')');
  %end
  drawnow;

  BACKCORR_DAT = [Wave; Time; min_nums; max_nums; min_numc; max_numc; 0; 0; ...
       ax_chro;Chro_line;Wave_field;Wave_slider;Ret_slider;ax_pca;pca_line;Contr];
```

```
    BACKCORR_DAT(16)=Contr;
    x=x;
elseif strcmp(action,'redraw'),
    Contr=BACKCORR_DAT(16);
    ax1=BACKCORR_DAT(14);
    line_handl1=BACKCORR_DAT(15);
    ax=BACKCORR_DAT(9);
    line_handle=BACKCORR_DAT(10);
    max_nums=BACKCORR_DAT(4);
    max_numc=BACKCORR_DAT(6);
    Wave=BACKCORR_DAT(1);
    Time=BACKCORR_DAT(2);
    Time1=Time+Contr-1;
    if Time1>=max_nums;
    Time1=max_nums;
    Time=max_nums-Contr+1;
    end
    t=1:max_nums;
    if (in1==1)
    set(BACKCORR_DAT(11),'string',num2str(Wave));
    set(BACKCORR_DAT(12),'value',Wave);
    if (ax~=gca),
    axes(ax);
    drawnow discard;
    end;
    %ax_chro=axes('Position',[.15 .40 .5 .26]);
    Chro_line=plot(t,x(:,Wave),'EraseMode','xor');
    %axis([-8 8 -.5 2]);
    grid;
    ylabel('A');
    xlabel('Retention Time');
    else
    %set(BACKCORR_DAT(13),'string',num2str(Time));
    %set(BACKCORR_DAT(14),'value',Time);
    if (ax1~=gca),
    axes(ax1);
    drawnow discard;
      end;
    [u,s,v]=svd(x(Time:Time1,:));
    Contr
    Time
    %max_numc
    %for i=1:Contr
```

```
%ax _pca=axes('Position',[.16 .24 .5 .4]);
pca _line=plot(1:max _numc,v(:,1:Contr),'EraseMode','xor');
%end

ylabel('A');
xlabel('Wavelength');
if Contr==2,
str1=['EV1: ',num2str(s(1,1))];
str2=[ EV2: ',num2str(s(2,2))];
str3=[          '];
elseif Contr>=3,
str1=[ EV1: ',num2str(s(1,1))];
str2=[ EV2: ',num2str(s(2,2))];
str3=[ EV3: ',num2str(s(3,3))];
end
str4=[ Region: ',num2str(Time),'-',num2str(Time+Contr-1)];
uicontrol('style','text','pos',[.70 .35 .28 .05],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String',str1);
uicontrol('style','text','pos',[.70 .28 .28 .05],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String',str2);
uicontrol('style','text','pos',[.70 .22 .28 .05],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String',str3);
uicontrol('style','text','pos',[.70 .42 .28 .05],...
    'Units','normalized','BackgroundColor','black',...
    'ForegroundColor','white','String',str4);
end

drawnow;

elseif strcmp(action,'ChanWav'),
% x=get(Wave _slider,'UserData')

if (in1==1), % set from slider
    BACKCORR _DAT(1)=floor(get(BACKCORR _DAT(12),'value'));

else % set from edit text
    min _numc=BACKCORR _DAT(5);
    max _numc=BACKCORR _DAT(6);
    Wave=floor(str2num(get(BACKCORR _DAT(11),'string')));
    if (Wave>max _numc-1),
        Wave=max _numc;
```

```
        end;
        if (Wave<min _ numc -1),
            Wave=min _ numc;
        end;
        BACKCORR _ DAT(1)=Wave;

    drawnow;
    end

elseif strcmp(action,'ChanRet'),
    % x =get('Wave _ slider,'UserData')

        min _ numc=BACKCORR _ DAT(5);

        BACKCORR _ DAT(2)=floor(get(BACKCORR _ DAT(13),'value'));
        max _ numc=BACKCORR _ DAT(6);
        Contr =BACKCORR _ DAT(16);
        Contr
    drawnow;

elseif strcmp(action,'done'),
    clf reset;
    clear global BACKCORR _ DAT
    delete(figure(1))
end

end
```

44. function name

函数 name 为一与函数 bakremo，函数 movpca 共同使用的程序，以保证全局变量的安全使用。使用该函数时，将要用的二维数据矩阵名以字符串输入即可。

【程序】

```
s=input('Please input the name of matrix in the form of "name"  ');save dataname s
```

45. function [scores，loads，ssq，res] = pca (data，plots，scl，lv)

函数 pca 用于主成分分析计算。

【例子】

[scores,loads,ssq,res] = pca(data,plots,scl,lv)

输入参数：

data 为二维数据矩阵；

plots 为一控制参数，

当 plots=0时，将抑制任何作图

当 plots=1时，对作图无限制

当 plots=2时，对作图有限制；

scl 为一选择性矢量参数，以对因子得分作图；

lv 为一选择性标量参数，表示引入计算的主成分数目。

输出参数：

scores：为主成分分析所得的因子得分矩阵；

loads：为主成分分析所得的因子载荷矩阵；

ssq：为方差信息；

res：为残差信息。

【程序】

```
if nargin < 2
  plots = 1;
end
if plots > 2
  error('Plot option must be 0, 1 or 2')
elseif plots < 0
  error('Plot option must be 0, 1 or 2')
end
[m,n] = size(data);
if n < m
  cov = (data' * data)/(m-1);
  [u,s,v] = svd(cov);
  temp2 = (1:n)';
  escl = 1:n;
else
  cov = (data * data')/(m-1);
  [u,s,v] = svd(cov);
  v = data' * v;
  for i = 1:m
     v(:,i) = v(:,i)/norm(v(:,i));
  end
  temp2 = (1:m)';
  escl = 1:m;
end
temp = diag(s) * 100/(sum(diag(s)));
ssq = [temp2 diag(s) temp cumsum(temp)];
% This section calculates the standard errors of the
% eigenvalues and plots them
if plots == 2
  eigmax = ssq(:,2)/(1-(1.96 * sqrt(2/m)));
  eigmin = ssq(:,2)/(1+(1.96 * sqrt(2/m)));
  clg
  plot(escl,ssq(:,2),escl,eigmax,'--b',escl,eigmin,'--b',escl,ssq(:,2),'og')
  title('Eigenvalue vs. PC Number showing 95% Confidence Limits')
  xlabel('PC Number')
  ylabel('Eigenvalue')
elseif plots == 1
  clg
```

```
    plot(escl,ssq(:,2),escl,ssq(:,2),'og')
    title('Eigenvalue vs. PC Number')
    xlabel('PC Number')
    ylabel('Eigenvalue')
end
% Print out the amount of variance captured
% disp('          ')
%disp(' Percent Variance Captured by PCA Model  ')
%disp('          ')
%disp(' PC#   Eigval   %Var   %TotVar')
%disp(ssq)
if nargin < 4
    input('How many principal components do you want to keep?   ');
    lv = ans;
end
if lv > n
    error('No. of PCs must be <= no. of variables')
end
if lv > m
    error('No. of PCs must be <= no. of samples')
end
% Form the PCA Model Based on the Number of PCs Chosen
loads = v(:,1:lv);
scores = data * loads;
I = eye(n);
% Calculate the standard error on the PC loadings if needed
if plots == 2
    loaderr = zeros(n,lv);
    if n > m, nn = m; else, nn = n; end
    for i = 1:lv
        dif = (ssq(:,2)-ones(nn,1) * ssq(i,2)).^2;
        dif = sort(dif);
        sig = sum((ones(nn-1,1) * ssq(i,2))./dif(2:nn,1));
        loaderr(:,i) = ((ssq(i,2)/m) * loads(:,i).^2) * sig;
    end
    loadmax = loads+loaderr;
    loadmin = loads-loaderr;
end
% Calculate the residuals matrix and Q values
%resmat = ((I-loads * loads') * data');
%res = (sum(resmat.^2))';
% Create the scale vectors to plot against
if plots >= 1.0
    if nargin < 3
```

```
    scl = 1:m;
    scllim = [1 m];
  else
    scllim = [min(scl) max(scl)];
  end
  scl2 = 1:n;
  temp = [1 1];
  for i = 1:lv
    pclim = sqrt(s(i,i)) * temp * 1.96;
    plot(scl,scores(:,i),scllim,pclim,'--b',scllim,-pclim,'--b')
    xlabel('Sample Number')
    str = sprintf('Score on PC# %g',i);
    ylabel(str)
    title('Sample Scores with 95% Limits')
    pause
    if plots == 2
plot(scl2,loads(:,i),scl2,loads(:,i),'og',scl2,loadmax(:,i),'--b',scl2,loadmin(:,i),'--b',[1 n],[0 0])
    elseif plots == 1
      plot(scl2,loads(:,i),scl2,loads(:,i),'og',[1 n],[0 0])
    end
  xlabel('Variable Number')
  str = sprintf('Loadings for PC# %g',i);
  ylabel(str)
  if plots ~= 2
    str = sprintf('Variable # vs. Loadings for PC# %g Showing Standard Errors',i);
    title(str)
  else
    str = sprintf('Variable Number vs. Loadings for PC# %g',i);
    title(str)
  end
  pause
end
end
% Calculate Q limit using unused eigenvalues
% * * * * * * HER STRYG ME AU
%temp = diag(s);
%if n < m
% emod = temp(lv+1:n,:);
%else
% emod = temp(lv+1:m,:);
%end
%th1 = sum(emod);
%th2 = sum(emod.^2);
%th3 = sum(emod.^3);
```

```
%h0 = 1-((2 * th1 * th3)/(3 * th2^2));
%if h0 <= 0.0
%h0 = .0001;
%disp('          ')
%disp('Warning: Distribution of unused eigenvalues indicates that')
%disp('         you should probably retain more PCs in the model.')
%end
%q = th1 * (((1.65 * sqrt(2 * th2 * h0^2)/th1) - 1 + th2 * h0 * (h0-1)/th1^2)^(1/h0));
%disp('          ')
%disp('The 95% Q limit is')
%disp(q)
if plots >= 1
  lim = [q q];
  plot(scl,res,scllim,lim,'--b')
  str = sprintf('Process Residual Q with 95 Percent Limit Based on %g PC Model', v);
  title(str)
  xlabel('Sample Number')
  ylabel('Residual')
  pause
end
%% Calculate T^2 limit using ftest routine
%**************
%if lv > 1
% if m > 300
%    tsq = (lv * (m-1)/(m-lv)) * ftest(.95,300,lv,2);
% else
%    tsq = (lv * (m-1)/(m-lv)) * ftest(.95,m-lv,lv,2);
% end
% disp('          ')
% disp('The 95% T^2 limit is')
% disp(tsq)
% Calculate the value of T^2 by normalizing the scores to
% unit variance and summing them up
  if plots >= 1.0
    temp2 = scores * inv(diag(ssq(1:lv,2).^.5));
    tsqvals = sum((temp2.^2)');
    tlim = [tsq tsq];
    plot(scl,tsqvals,scllim,tlim,'--b')
    str = sprintf('Value of T^2 with 95 Percent Limit Based on %g PC Model',lv);
    title(str)
    xlabel('Sample Number')
    ylabel('Value of T^2')
  end
 %else
```

```
% disp('T^2 not calculated when number of latent variables = 1')
  tsq = 1.96^2;
end
```

46. function [C,S]=rescomp(X,indices);

函数 rescomp 为在每个化学组分都有选择性信息时的分辨程序。

输入参数：

x 为1个二维数据矩阵，该矩阵的每一行为一个光谱，每一列为一个色谱（或一滴定曲线，或一动力学系统的连续时间记录，等）；

indices 为包含各组分的选择性区域的起点和终点的矩阵，例

如对两组分体系有 indices= [14 18 43 49]。

输出参数：

C 为包含二维数据阵中的各组分浓度的矩阵；

S 为包含二维数据阵中的各组分纯光谱的矩阵。

【程序】

```
numc=size(indices,1);

for i=1:numc
    [t,p]=pca(X(indices(i,1):indices(i,2),:),0,(1:2),1);
    S=[S p];
end
C= X * S * inv(S' * S);
```

47. function [C,S,Xred]=respart(X,indices,numc);

函数 respart 为在部分化学组分有选择性信息时的分辨程序。

输入参数：

x 为1个二维数据矩阵，该矩阵的每一行为一个光谱；每一列为一个色谱（或一滴定曲线，或一动力学系统的连续时间记录，等）；

indices 为包含各组分的选择性区域的起点和终点、及各组分的零浓度（包括左零浓度、和右零浓度）的矩阵，例：

如对四组分体系，其中有两个组分有选择性信息，则有

indices= [14 18 1 12 56 80 43 49 1 18 63 80]

其中14和18为一个组分的选择性区域的起点与终点，1、12和56、80分别为其左零浓度和右零浓度区域。第二行各数值的意义同此。

注意：当某一组分只有一个零浓度区域时，另一个区域可用两个0替代。

numc 为被分辨体系中所包含的化学组分数目。

输出参数：

C 为包含二维数据阵中的各组分浓度的矩阵。

S 为包含二维数据阵中的各组分纯光谱的矩阵。

Xred 为残差数据矩阵。注意，此时所得的残差数据矩阵还含有别的未被分辨组分的信息。

【程序】

```
[n,m]=size(X);
numa=size(indices,1);

[T,P]=pca(X,0,(1:2),numc);
```

```
for i=1:numa

    [tc,pc]=pca(X(indices(i,1):indices(i,2),:),0,(1:2),1);
    T2=zeros(n,numc);
    c=zeros(n,1);
    k=1;

    for j=indices(i,1):indices(i,2)
        T2(j,:)=T(j,:);
        c(j)=tc(k);
        k=k+1;
    end

    for j=indices(i,3):indices(i,4)
        T2(j,:)=T(j,:);
    end

    if indices(i,5) ~= 0

        for j=indices(i,5):indices(i,6)
            T2(j,:)=T(j,:);
        end
    end

    r=inv(T2' * T2) * T2' * c;
    cpure=T * r;
    C=[C cpure];
    S=[S pc];
end

Xred=X-C * S';
```

48. function indices=scoop (X);

函数 scoop 可用于通过特征投影图来找到组分的选择性信息。这样得到的信息可用于下一步的分辨。

输入参数：

x 为1个二维数据矩阵，该矩阵的每一行为一个光谱；每一列为一个色谱（或一滴定曲线，或一动力学系统的连续时间记录，等）。

输出参数：indices 为包含各组分的选择性区域的起点和终点的矩阵。

【程序】

```
k=1;

[n,m]=size(X);
[T,P]=pca(X,0,(1:2),2);

plot(T(:,1),T(:,2))
```

```
hold on
plot(T(:,1),T(:,2),'o')
xlabel('PC1')
ylabel('PC2')
zoom
slutt=1;

while(slutt==1)
    disp('Use the mouse to enter the start of the region of interest')
    [x,y]=ginput(1);
    dist=1e8;
    for i=1:n
        newdist=norm([x y]-[T(i,1) T(i,2)]);
        if newdist<dist
            dist=newdist;
            ind1=i;
        end
    end

    disp('Use the mouse to enter the last point of the region of interest')
    [x,y]=ginput(1);
    dist=1e8;
    for j=1:n
        newdist=norm([x y]-[T(j,1) T(j,2)]);
        if newdist<dist
            dist=newdist;
            ind2=j;
        end
    end

    if ind2<ind1
        q=ind1;
        ind1=ind2;
        ind2=q;
    end

    d1=norm([T(ind1,1) T(ind1,2)]-[0 0]);
    d2=norm([T(ind2,1) T(ind2,2)]-[0 0]);

    if d1>d2
        vec1=[0;T(ind1,1)];
        vec2=[0;T(ind1,2)];
    else
        vec1=[0;T(ind2,1)];
```

```
        vec2 = [0;T(ind2,2)];
    end

    linje=plot(vec1,vec2,'r');
    p1=plot(T(ind1,1),T(ind1,2),'g+');
    p2=plot(T(ind2,1),T(ind2,2),'g*');

    answer=input('True selective region? (0 = No, 1 = Yes )');

    if answer == 1
        indices(k,1)=ind1;
        indices(k,2)=ind2;
        k=k+1;
    end

    delete(linje);
    delete(p1);
    delete(p2);

    slutt=input('Do you want to try to find another selective region? (0= No,1=Yes) ');
end
hold off
```

49. function y=selnorm (x, thres);

函数 selnorm 用于对数据矩阵进行选择性标准化，以消除不等性方差的对体系组分数估计的影响。

输入参数：

x 为1个二维数据矩阵，该矩阵的每一行为一个光谱；每一列为一个色谱（或一滴定曲线，或一动力学系统的连续时间记录，等）；

thres 为标准化阈值。

输出参数：

y 为经选择性标准化后得到的数据矩阵。可直接用于函数 seqrank 的特征结构跟踪分析，以得到化学秩图

【程序】

```
[m,n]=size(x);
for i=1:m
    if sum(x(i,:))>=thres
        y(i,:)=x(i,:)./sum(x(i,:))*thres;
    else
        y(i,:)=x(i,:);
    end
end
```

50. function indices =seqrank (X, noiselimit);

函数 seqrank 可用于特征结构跟踪分析，以得到保留时间方向的化学秩图。该程序据供了一个图界面，用户就直接用鼠标在图上将可能的选择性区域点出，程序就可对该区域进行计算，并通过对话，要用户确定它是否是选择性区域，一经确定，程序将自动将该区域计下。

输入参数：

x 为1个二维数据矩阵，该矩阵的每一行为一个光谱；每一列为一个色谱（或一滴定曲线，或一动力学系统的连续时间记录，等）；

noiselimit 为自定义噪声水平。

输出参数：

indices 为包含各组分的选择性区域的起点和终点的矩阵。

【程序】

```
[n,m]=size(X);
winsize=input('Enter the size of the moving window ');
j=1;
slutt=1;
color=['ymcrgbw'];
col=1;
mx=mean(X');

for i=1:n-winsize+1
    [T,P]=pca(X(i:i+winsize-1,:),0,(1:2),winsize);
    E=T' * T;
     eigs(j,:)=(diag(E))';
    j=j+1;
end
eigs=log10(eigs);

no=ones(1,n);
no=no * noiselimit;
no=log10(no);
vec=(1:n);

j=1;

subplot(211),plot(mx)
hold on
subplot(212),plot(eigs)
hold on
linje=subplot(212),plot(vec,no,'w');;
zoom

while slutt == 1

    disp('Use the mouse pointer to mark the suspected selective regions')
    disp('Mark the lower index first')
    [x,y]=ginput(2);
    low=floor(x(1));
    if low < 1
```

```
        low=1;
    end
    high=ceil(x(2));
    if high > n
        high=n;
    end

    [t,p]=pca(X(low:high,:),0,(1:2),2);

    figure(2)
    plot(p)
    E=t'*t;
    title(['The first eigenvalue is ',num2str(E(1,1))])
    xlabel(['Second ',num2str(E(2,2)),' Noise:  ',num2str(noiselimit)])

    answer=input('True selective region? (1=Yes, 0=No)  ');
    if answer == 1
        figure(1)
        subplot(211),bar(low:high,mx(low:high),color(col))
        col=col+1;
        if col > 7
            col=1;
        end
        selinc(j,:)=[low high];
        j=j+1;
    end
    delete(figure(2))
    slutt=input('Do you want to continue testing? (1=Yes, 0=No)  ');
end
```

51. function [simu, y, y1] =Simu2D

函数 Simu2D 为产生二维数据的计算机模拟程序。最高可产生包含多组分混合体系。本程序采用对话形式来输入模拟所需参数。每个光谱由两个高斯峰所组成，每个色谱由一个高斯峰所组成。

输出参数

simu 为所得的二维（100×100）阶的数据矩阵；

y 为模拟纯组分光谱矩阵；

y1为模拟纯组分色谱矩阵。

【程序】

```
% This is a program to produce the simulation two-way data. The size
% of the data matrix is 100*100 (simu). The chemical components in the simulated
% system can be up to six. The pure spectra of each component consist
% of two Gauss peaks, while the pure chromatographies of each component
% consist of one Gauss peak. The parameters of the Gauss peaks should.
% be, therefore, input by the user.
%
```

```
% y=C;y1=S;
% Dept. of Chemistry, Hunan University. PRC.
% By Yizeng Liang, 14/8, 1994

ComNum=input('Please input the number of components in your simulation system:  ');
disp('          ')
disp('Now, you are asked to input the parameters of pure spectra!')

for i=1:ComNum
Spe(i,1)=input('Please input the peak position of the first Gauss peak(1-100):  ');
Spe(i,2)=input('Please input the peak position of the second Gauss peak(1-100):  ');
Spe(i,3)=input('Please input the standard deviation of the first Gauss peak(1-30):  ');
Spe(i,4)=input('Please input the standard deviation of the second Gauss peak(1-30):  ');
end
disp('          ')
disp('Now, you are asked to input the parameters of pure Chromatographies!');
disp('          ')
for i=1:ComNum
Chr(i,1)=input('Please input the peak position of the Gauss peak(1--100):  ');
Chr(i,2)=input('Please input the standard deviation of the Gauss peak(1--30):  ');
end
disp('Now, you are asked to input the heights of pure Chromatographies!(0-1.0)');
disp('          ')
for i=1:ComNum
mul(i)=input('Please input the height of the Gauss peak(0--1):          ');
end
for i=1:ComNum
     for j=1:100
          y(j,i)=mul(i)*exp(-(j-Chr(i,1))^2/(Chr(i,2)^2));
     end
end
for i=1:ComNum
     for j=1:100
          y1(j,i)=exp(-(j-Spe(i,1))^2/(Spe(i,3)^2*2))+.7*exp(-(j-Spe(i,2))^2/(Spe(i,4)^
2*2));
     end
   end
   simu=zeros(100);
   simu=y*y1';
simu=simu+randn([100,100])*.0001;
figure(1)
subplot(221),plot(1:100,y)
xlabel('Retention Time')
subplot(222),plot(1:100,y1)
```

```
xlabel('Wavelength')
if ComNum<=3
plo=100 * ones(100,1);
subplot(212),plot([0 100 0 100],[0 100 100 0],'o')
s2='Sp1 Sp2 Sd1 Sd2 Cp Cd';
text(50,80,s2);
for i=1:ComNum
s1='Parameter for component';
s1 =[s1,'     ',num2str(i),':     ',num2str(Spe(i,1)),'     ',num2str(Spe(i,2)),'     ',num2str(Spe(i,
3)),'     ',num2str(Spe(i,4)),'     ',num2str(Chr(i,1)),'     ',num2str(Chr(i,2))];
text(2,15 * (5-i),s1);
end
else
plo=100 * ones(100,1);
subplot(212),plot([0 100 0 100],[0 100 100 0],'o')
s2='Sp1 Sp2 Sd1 Sd2 Cp Cd';
text(50,80,s2);
for i=1:ComNum
s1='Parameter for component';
s1 =[s1,'     ',num2str(i),':     ',num2str(Spe(i,1)),'     ',num2str(Spe(i,2)),'     ',num2str(Spe(i,
3)),'     ',num2str(Spe(i,4)),'     ',num2str(Chr(i,1)),'     ',num2str(Chr(i,2))];
text(10,10 * (8-i),s1);
end
end
pause(10)
figure(2)
mesh(simu)
xlabel('Wavelength')
ylabel('Retention Time')
end
```

52. % Zoom 2D xy

函数 Zoom 为作图缩放功能函数。

【程序】

```
AAAz=[...
  '[AAAr,AAAk]=ginput(2); '...
  'axis([min(AAAr) max(AAAr) min(AAAk) max(AAAk)]); '...
  'clear AAAr AAAk ; '];
AAAe='clf reset; clear AAAz AAAe AAAa AAAzh AAAah AAAch ;';
AAAa='axis(''auto'');';

AAAah = uicontrol('style','push','units','normal',...
            'pos',[0 .80 .08 .09],'string','Auto','call',AAAa);

AAAzh = uicontrol('style','push','units','normal',...
```

```
        'pos',[0 .90 .08 .09],'string','Zoom','call',AAAz);

AAAeh = uicontrol('style','push','units','normal',...
        'pos',[0 .70 .08 .09],'string','End','call',AAAe);
```

53. function class = classify (sample, training, group)

函数 CLASSIFY 为线性判别分析子程序。

【例子】

class=CLASSIFY (SAMPLE, TRAINING, GROUP)

输入参数：

SAMPLE 为待分类的样本矩阵；

TRAINING 为已知类别的训练矩阵；

GROUP 为已知类别编号的训练矢量。

输出参数：class 对应于 SAMPLE 的每个样本的类别。

【程序】

```
[gr, gc] = size(group);
if min(gr,gc) ~= 1
  error ('Requires a vector third argument.');
end

if gc ~= 1,
  group = group (:);
  gr = gc;
end

if any (group-round(group)) | any(group) < 1
  error ('The third input argument must be positive integers.');
end
maxg = max(group);

[tr,tc] = size(training);
if tr ~= gr,
  error ('The number of rows in the second and third input arguments must match.');
end

[sr, sc] = size (sample);
if sc ~= tc
  error ('The number of columns in the first and second input arguments must match.');
end

d = zeros(sr,maxg);
for k = 1: maxg
  groupk = training (find(group == k),:);
  d(:,k) = mahal(sample,groupk);
end
```

```
[tmp, class] = min(d');
class = class';
```

54. function d = mahal(Y, X);

函数 MAHAL 为计算 Mahalanobis 距离的子程序。

【例子】

d = MAHAL (Y, X)

输入参数：

Y 为样本矩阵或矢量；

X 为已知类别的训练矩阵。

输出参数：d 为它们之间的距离。

【程序】

```
[rx,cx] = size(X);
[ry,cy] = size(Y);

if cx ~= cy
  error('Requires the inputs to have the same number of columns.');
end

if rx < cx
  error('The number of rows of X must exceed the number of columns.');
end

m = mean(X);
C = X-m(ones(rx,1),:);
[Q,R] = qr(C,0);
R = R./sqrt(rx-1);
ri = R\eye(cx,cx);

M = m(ones(ry,1),:);
E = ((Y-M) * ri)';
d = sum(E.* E)';
```

55. function [center, U, obj _ fcn] = fcm (data, cluster _ n, options)

函数 FCM 为采用模糊 K _均值聚类方法对样本矩阵进行聚类子程序。

【例子】

(1) [CENTER, U, OBJ _FCN] = FCM(DATA, CLUSTER _N)

输入参数：

DATA 为待分类的样本矩阵；

CLUSTER _N 为待分类数目（一般大于1）。

输出参数：

CENTER 计算所得的类中心矢量；

U 为计算所得的最终模糊分配矩阵 (fuzzy partition matrix or MF matrix)；

OBJ _ FCN 为在迭代过程中的目标函数值。

(2) [CENTER，U，OBJ_FCN] = FCM(DATA，CLUSTER_N，OPTIONS)

输入参数：

DATA 为待分类的样本矩阵；

CLUSTER_N 为待分类数目（一般大于1）；

OPTIONS 为一控制聚类方法进行的选择矢量，其中，OPTIONS（1）为分配矩阵的指数（exponent）（预置值为：2.0）

OPTIONS（2）：最大迭代数目（预置值为：100）

OPTIONS（3）：迭代改进的最小量（预置值为：1e-5）

OPTIONS（4）：迭代信息显示的控制参数（预置值为：1）。

输出参数：

CENTER 计算所得的类中心矢量；

U 为计算所得的最终模糊分配矩阵；

OBJ_FCN 为在迭代过程中的目标函数值。

使用示例：

```
» data = rand(100, 2);
» [center, U, obj_fcn] = fcm(data, 2);
» plot(data(:, 1), data(:, 2), 'o');
» maxU = max(U);
» index1 = find(U(1, :) == maxU);
» index2 = find(U(2, :) == maxU);
» line(data(index1, 1), data(index1, 2), 'linestyle', '*', 'color', 'g');
» line(data(index2, 1), data(index2, 2), 'linestyle', '*', 'color', 'r');
```

参见 INITFCM，DISTFCM，STEPFCM.

【程序】

```
if nargin ~= 2 & nargin ~= 3,
  error('Too many or too few input arguments!');
end

data_n = size(data, 1);
in_n = size(data, 2);

% Change the following to set default options
default_options = [2;   % exponent for the partition matrix U
        100;   % max. number of iteration
        1e-5;   % min. amount of improvement
        1];   % info display during iteration

if nargin == 2,
    options = default_options;
else
    % If "options" is not fully specified, pad it with default values.
    if length(options) < 4,
      tmp = default_options;
      tmp(1:length(options)) = options;
```

```
        options = tmp;
    end
    % If some entries of "options" are nan's, replace them with defaults.
    nan_index = find(isnan(options)==1);
    options(nan_index) = default_options(nan_index);
    if options(1) <= 1,
        error('The exponent should be greater than 1!');
    end
end

expo = options(1);      % Exponent for U
max_iter = options(2);      % Max. iteration
min_impro = options(3);     % Min. improvement
display = options(4);    % Display info or not

obj_fcn = zeros(max_iter, 1);   % Array for objective function

U = initfcm(cluster_n, data_n);    % Initial fuzzy partition
% Main loop
for i = 1:max_iter,
    [U, center, obj_fcn(i)] = stepfcm(data, U, cluster_n, expo);
    if display,
        fprintf('Iteration count = %d, obj. fcn = %f\n', i, obj_fcn(i));
    end
    % check termination condition
    if i > 1,
        if abs(obj_fcn(i)-obj_fcn(i-1)) < min_impro, break; end,
    end
end

iter_n = i;   % Actual number of iterations
obj_fcn(iter_n+1:max_iter) = [];
```

56. function U = initfcm (cluster _ n, data _ n)

函数 INITFCM 产生模糊 K _均值聚类方法的初始分配矩阵子程序。

【例子】

U = INITFCM (CLUSTER _ N, DATA _ N)

输入参数：

CLUSTER _ N 为需分类的数目；

DATA _ N 样本数据矢量数目。

输出参数：U 为随机产生的模糊分配矩阵，其大小为（CLUSTER _ N×DATA _ N），其每一列的和为1。

参见 DISTFCM，STEPFCM，FCM.

【程序】

```
U = rand (cluster_n, data_n);
```

```
col_sum = sum (U);
U = U./col_sum (ones (cluster_n, 1), :);
```

57. function [U_new, center, obj_fcn] = stepfcm (data, U, cluster_n, expo)

函数 STEPFCM 为模糊 K_均值聚类方法中迭代一步计算子程序。

【例子】

[U_NEW, CENTER, ERR] = STEPFCM(DATA, U, CLUSTER_N, EXPO)

输入参数：

DATA 为待分类的样本数据矩阵；

U：模糊分配矩阵；

CLUSTER_N：需分类的数目；

EXPO：模糊分配矩阵指数（> 1）。

输出参数：

CENTER：类中心矢量；

ERR：模糊分类的目标函数值。

参见　DISTFCM, INITFCM, FCM.

【程序】

```
mf = U.^expo;        % MF matrix after exponential modification
center = mf * data./((ones (size (data, 2), 1) * sum (mf'))'); % new center
dist = distfcm (center, data);        % fill the distance matrix
obj_fcn = sum (sum ((dist.^2) . * mf));       % objective function
tmp = dist.^ (-2/ (expo-1));       % calculate new U, suppose expo != 1
U_new = tmp./ (ones (cluster_n, 1) * sum (tmp));
```

58. function out = distfcm (center, data)

函数 DISTFCM 可用于模糊 K_均值聚类方法中计算数据矩阵中样本点到类中心的欧氏（Euclidean）距离。

【例子】

OUT = DISTFCM (CENTER, DATA)

输入参数：

DATA 为样本矩阵；

CENTER 为类中心矢量。

输出参数：OUT 为计算所得距离，如 OUT (I, J) 为 CENTER (I,:) 与 DATA (J,:) 之间的距离。

参见 INITFCM, STEPFCM, FCM.

【程序】

```
out = zeros(size(center, 1), size(data, 1));

% fill the output matrix

if size(center, 2) > 1,
    for k = 1:size(center, 1),
    out(k, :) = sqrt(sum(((data-ones(size(data, 1), 1) * center(k, :)).^2)'));
    end
else      % 1--D data
    for k = 1:size(center, 1),
    out(k, :) = abs(center(k)-data)';
```

```
    end
end
```

59. function [timeseri, y] =Rung _ kuta (x0, time _ range, stepwise);

函数 Rung _ kuta 为四阶 Runge-Kutta 法;

【例子】

[timeseri, y] =Rung _ kuta (x0, time _ range, stepwise);

输入参数:

x0 为常微分组的初始值矢量;

time _ range 为积分计算区间矢量;

stepwise 为积分步长。

输出参数:

timeseri 积分时间序列;

y 为在积分时间序列的常微分组的解矩阵。

【程序】

```
[m,l]=size(x0);
timeseri=[time _ range(1):stepwise:time _ range(2)];
n=ceil((time _ range(2)-time _ range(1))/stepwise);

y(:,1)=x0;

for i=1:n
    k(:,1)=stepwise * dery(y(:,i));
    k(:,2)=stepwise * dery(y(:,i)+stepwise/2 * k(:,1));
    k(:,3)=stepwise * dery(y(:,i)+stepwise/2 * k(:,2));
    k(:,4)=stepwise * dery(y(:,i)+stepwise * k(:,3));
    y(:,i+1)=y(:,i)+(k(:,1)+2 * k(:,2)+2 * k(:,3)+k(:,4))/6;
end
size(timeseri)
size(y)
pause
plot(timeseri',y');
```

60. function [timeseri,y]=Adambm(x0,time _ range,stepwise);

函数 Adambm 为求解为常微分组的初值问题的 Adams 预估_校正方法;

【例子】

[timeseri, y] =Adambm (x0, time _ range, stepwise);

输入参数:

x0 为常微分组的初始值矢量;

time _ range 为积分计算区间矢量;

stepwise 为积分步长。

输出参数:

timeseri 积分时间序列;

y 为在积分时间序列的常微分组的解矩阵。

【程序】

```
y(:,1)=x0
```

```
timeseri=[time_range(1):stepwise:time_range(2)];
n=ceil((time_range(2)-time_range(1))/stepwise);
[tim,y]=Rung_kuta(x0,[time_range(1) time_range(1)+4*stepwise],stepwise);
for i=4:n
  y(:,i+1)=y(:,i)+stepwise*(2.291666666*dery(y(:,i))-2.4583333*dery(y(:,i-1))...
           +1.541666666*dery(y(:,i-2))-0.375*dery(y(:,i-3)));
  y(:,i+1)=y(:,i)+stepwise*(0.375*dery(y(:,i+1))+0.79166666*dery(y(:,i))...
           -0.2083333333*dery(y(:,i-1))+0.04166666666*dery(y(:,i-2)));
end
size(timeseri)
size(y)
pause
plot(timeseri',y');
```

61. function [timeseri,y]=Hamming(x0,time_range,stepwise);

函数 Hamming 为求解为常微分组的初值问题的 Hamming 定步长方法；

【例子】

[timeseri,y]=Hamming(x0,time_range,stepwise);

输入参数：

x0为常微分组的初始值矢量；

time_range 为积分计算区间矢量；

stepwise 为积分步长。

输出参数：

timeseri 积分时间序列；

y 为在积分时间序列的常微分组的解矩阵。

【程序】

```
y(:,1)=x0
timeseri=[time_range(1):stepwise:time_range(2)];
n=ceil((time_range(2)-time_range(1))/stepwise);
[tim,y]=Rung_kuta(x0,[time_range(1) time_range(1)+5*stepwise],stepwise);
r(:,1)=[1 0]';
for i=5:n
  c(:,1)=(9*y(:,i-1)-y(:,i-3)+3*stepwise*(dery(r(:,1))+...
          2*dery(y(:,i-1))-dery(y(:,i-2))))/8;
  p(:,1)=y(:,i-4)+8*stepwise*(dery(y(:,i-1))+dery(y(:,i-3))...
         -0.5*dery(y(:,i-2)))/3;
  p(:,2)=y(:,i-3)+8*stepwise*(dery(y(:,i))+dery(y(:,i-2))...
         -0.5*dery(y(:,i-1)))/3;
  r(:,2)=p(:,2)-112/121*(p(:,1)-c(:,1));
  c(:,2)=(9*y(:,i)-y(:,i-2)+3*stepwise*(dery(r(:,2))+2*dery(y(:,i))...
          -dery(y(:,i-1))))/8;
  y(:,i+1)=c(:,2)+9*(p(:,2)-c(:,2))/121;
  c(:,1)=c(:,2);
  p(:,1)=p(:,2);
  r(:,1)=r(:,2);
```

```
end
size(timeseri)
size(y)
pause
plot(timeseri',y');
```

62. function f=dery (y)

函数 dery 为定义一个常微分方程组的调用函数。

该函数定义了下述微分方程：

dy1/dt=y2

dy2/dt=-y1

所以，根据不同的函数，应有不同的 dery 函数。

【程序】

```
[m,n]=size(y);
f(1)=y(2);
f(2)=-y(1);
f=f';
```

63. function [tout,yout,o3,o4,o5,o6] = ode45(odefile,tspan,y0,options,varargin)

函数 ODE45为解决非刚性常微分方程的初值问题的通用程序。

【例子】

(1) [T,Y] = ODE45('F',TSPAN,Y0)

输入参数：

TSPAN 为积分计算区间矢量，如 TSPAN = [T0 TFINAL]；

Y0为常微分组的初始值矢量；

字符串参数 F 为定义一个常微分方程组的调用函数。

输出参数：

T 积分时间序列；

Y 为在积分时间序列的常微分组的解矩阵。

使用示例：

1) 先用 M 文件定义一个常微分方程组的调用函数 vdpol (t, x)：

```
function xdot=vdpol(t,x);
xdot(1)=x(1)*(1-x(2).^2)-x(2);
xdot(2)=x(1);
xdot=xdot';
```

将此函数存为 M 文件。

2) 继用下述步骤来调用函数 ode45，并将结果在屏幕上显示：

```
>t0=0;
>tf=20;
>x0=[0.25]
>[t,x]=ode45('vdpol',[t0 tf],x0');
>plot(t,x)
```

(2) [T,Y] = ODE45('F',TSPAN,Y0,OPTIONS)

输入参数：

TSPAN 为积分计算区间矢量，如 TSPAN = [T0 TFINAL]；

Y0为常微分组的初始值矢量；

字符串参数 F 为定义一个常微分方程组的调用函数；

OPTIONS 为选择参数集。

输出参数：

T 积分时间序列；

Y 为在积分时间序列的常微分组的解矩阵。

使用示例：

1) OPTIONS 可由下述函数调用得到：

```
) OPTIONS = odeset('RelTol',1e-4,'AbsTol',[1e-4 1e-4 1e-5]);
```

在此：字符串参数'RelTol'表示相对误差要求(relative error tolerance)，赋值为1e—4；

字符串参数'AbsTol'表示绝对误差要求 (absolute error tolerance)，对一个三阶常微分组中的三个数值积分解分别赋值为 [1e—4 1e—4 1e—5]。

2) 用 M 文件定义一个常微分方程组的调用函数 vdpol (t，x)：

```
function xdot= rigidode(t,x);
xdot(1)=x(1)*(1-x(2).^2)-x(2);
xdot(2)=x(1);
xdot(3)=-x(2);
xdot=xdot';
```

将此函数存为 M 文件。

3) 继用下述步骤来调用函数 ode45，并将结果在屏幕上显示：

```
>t0=0;
>tf=20;
>x0=[0.25]
>[t,x]=ode45('rigidode',[t0 tf],x0,OPTIONS);
>plot(t,x)
```

【程序】

```
true = logical(1);
false = ~true;

nsteps = 0;              % stats
nfailed = 0;             % stats
nfevals = 0;             % stats
npds = 0;                % stats
ndecomps = 0;            % stats
nsolves = 0;             % stats

if nargin == 0
  error('Not enough input arguments. See ODE45.');
elseif ~isstr(odefile) & ~isa(odefile, 'inline')
  error('First argument must be a single-quoted string. See ODE45.');
end

if nargin == 1
  tspan = []; y0 = []; options = [];
elseif nargin == 2
```

```
  y0 = []; options = [];
elseif nargin == 3
  options = [];
elseif ~isempty(options) & ~isa(options,'struct')
  if (length(tspan) == 1) & (length(y0) == 1) & (min(size(options)) == 1)
    tspan = [tspan; y0];
    y0 = options;
    options = [];
    varargin = {};
    msg = sprintf('Use ode45("%s",tspan,y0,...) instead.',odefile);
    warning(['Obsolete syntax. ' msg]);
  else
    error('Correct syntax is ode45("odefile",tspan,y0,options).');
  end
end

if isstr(odefile) & (exist(odefile) == 4) % a SIMULINK model
  warning(['Future versions of this M-file will not support '...
          'SIMULINK models. Use the SIM command directly.']);

  options=simset('Solver','ode45',...
                 'InitialState', y0,...
                 'RelTol',odeget(options,'RelTol'),...
                 'AbsTol',odeget(options,'AbsTol'),...
                 'Refine',odeget(options,'Refine'),...
                 'MaxStep',odeget(options,'MaxStep'),...
                 'InitialStep',odeget(options,'InitialStep'));

  % The code before and after the call to sim copies the ToWorkspace
  %     variables to the caller workspace.
  % Note: we must use unique variables names at this point to be
  %     distinct from ToWorkspace variable names.
  odefile_cp2wksp = odefile;
  [tout_cp2wksp,yout_cp2wksp,o3_cp2wksp] = ...
                sim(odefile,tspan,options,varargin{:});

  % Copy ToWorkspace variables to the caller workspace.
  names_cp2wksp =
 find_system(odefile_cp2wksp,'LookUnderMasks','on',...
                  'BlockType', 'ToWorkspace');
  for i_cp2wksp = 1:length(names_cp2wksp)
    varname_cp2wksp = get_param(names_cp2wksp{i_cp2wksp},...
                               'VariableName');
    assignin('caller', varname_cp2wksp, eval(varname_cp2wksp));
```

```
    end
    % Now safe to use tout, yout, and o3
    tout = tout_cp2wksp; yout = yout_cp2wksp; o3 = o3_cp2wksp;

    return;
  end

% Get default tspan and y0 from odefile if none are specified.
if isempty(tspan) | isempty(y0)
  if (nargout(odefile) < 3) & (nargout(odefile) ~= -1)
    msg = sprintf('Use ode45("%s",tspan,y0,...) instead.',odefile);
    error(['No default parameters in ' upper(odefile) '. ' msg]);
  end
  [def_tspan,def_y0,def_options] = feval(odefile,[],[],'init',varargin{:});
  if isempty(tspan)
    tspan = def_tspan;
  end
  if isempty(y0)
    y0 = def_y0;
  end
  if isempty(options)
    options = def_options;
  else
    options = odeset(def_options,options);
  end
end

% Test that tspan is internally consistent.
tspan = tspan(:);
ntspan = length(tspan);
if ntspan == 1
  t0 = 0;
  next = 1;
else
  t0 = tspan(1);
  next = 2;
end
tfinal = tspan(ntspan);
if t0 == tfinal
  error('The last entry in tspan must be different from the first entry.');
end
tdir = sign(tfinal - t0);
if any(tdir * (tspan(2:ntspan)-tspan(1:ntspan-1)) <= 0)
  error('The entries in tspan must strictly increase or decrease.');
```

```
end

t = t0;
y = y0(:);
neq = length(y);

% Get options. and set defaults.
rtol = odeget(options,'RelTol',1e-3);
if (length(rtol) ~= 1) | (rtol <= 0)
  error('RelTol must be a positive scalar.');
end
if rtol < 100 * eps
  rtol = 100 * eps;
  warning(['RelTol has been increased to ' num2str(rtol) '.']);
end

atol = odeget(options,'AbsTol',1e-6);
if any(atol <= 0)
  error('AbsTol must be positive.');
end

normcontrol = strcmp(odeget(options,'NormControl','off'),'on');
if normcontrol
  if length(atol) ~= 1
    error('Solving with NormControl "on" requires a scalar AbsTol.');
  end
  normy = norm(y);
else
  if (length(atol) ~= 1) & (length(atol) ~= neq)
    error(sprintf (['Solving %s requires a scalar AbsTol, '...
                    'or a vector AbsTol of length %d'],upper(odefile),neq));
  end
  atol = atol(:);
end
threshold = atol / rtol;

% By default, hmax is 1/10 of the interval.
hmax = min(abs(tfinal-t), abs(odeget(options,'MaxStep',0.1 * (tfinal-t))));
if hmax <= )
  error('Option "MaxStep" must be greater than zero.');
end
htry = abs(odeget(options,'InitialStep'));
if htry <= 0
  error('Option "InitialStep" must be greater than zero.');
```

```
end

haveeventfun = strcmp(odeget(options,'Events','off'),'on');
if haveeventfun
  valt = feval(odefile,t,y,'events',varargin{:});
  teout = [];
  yeout = [];
  ieout = [];
end

if nargout > 0
  outfun = odeget(options,'OutputFcn');
else
  outfun = odeget(options,'OutputFcn','odeplot');
end
if isempty(outfun)
  haveoutfun = false;
else
  haveoutfun = true;
  outputs = odeget(options,'OutputSel',1:neq);
end
refine = odeget(options,'Refine',4);      % Necessary for smooth plots.
printstats = strcmp(odeget(options,'Stats','off'),'on');

if strcmp(odeget(options,'Mass','off'),'on')
  error('Solver does not handle mass matrices, M * y'        '. See ODE15S or ODE23S.');
end

% Set the output flag.
if ntspan > 2
  outflag = 1;                  % output only at tspan points
elseif refine <= 1
  outflag = 2;                  % computed points, no refinement
else
  outflag = 3;                  % computed points, with refinement
  S = (1:refine-1)' / refine;
end

% Allocate memory if we're generating output.
if nargout > 0
  if ntspan > 2                 % output only at tspan points
    tout = zeros(ntspan,1);
    yout = zeros(ntspan,neq);
  else                          % alloc in chunks
```

```
    chunk = max(ceil(128 / neq),refine);
    tout = zeros(chunk,1);
    yout = zeros(chunk,neq);
  end
  nout = 1;
  tout(nout) = t;
  yout(nout,:) = y.';
end

% Initialize method parameters.
pow = 1/5;
A = [1/5; 3/10; 4/5; 8/9; 1; 1];
B = [
    1/5     3/40    44/45    19372/6561     9017/3168      35/384
    0       9/40    -56/15   -25360/2187    -355/33        0
    0       0       32/9     64448/6561     46732/5247     500/1113
    0       0       0        -212/729       49/176         125/192
    0       0       0        0              -5103/18656    -2187/6784
    0       0       0        0              0              11/84
    0       0       0        0              0              0
    ];
E = [71/57600; 0; -71/16695; 71/1920; -17253/339200; 22/525; -1/40];
f = zeros(neq,7);

% Maintain backward compatibility with ODE files that don't accept flags.
if isempty(varargin)
  args = {};
else                            % There are extra arguments, so
  args = [{[]} varargin];       % stick an empty flag on the front
end                             % before sending them to odefile.

f0 = feval(odefile,t,y,args{:});
nfevals = nfevals + 1;                % stats
[m,n] = size(f0);
if n > 1
  error([upper(odefile) ' must return a column vector.'])
elseif m ~= neq
  msg = sprintf('an initial condition vector of length %d.',m);
  error(['Solving ' uppar(odefile) ' requires ' msg]);
end

hmin = 16 * eps * abs(t);
if isempty(htry)
  % Compute an initial step size h using y'(t).
```

```
    absh = min(hmax, abs(tspan(next) - t));
    if normcontrol
      rh = (norm(f0) / max(normy,threshold)) / (0.8 * rtol ^ pow);
    else
      rh = norm(f0 ./ max(abs(y),threshold),inf) / (0.8 * rtol ^ pow);
    end
    if absh * rh > 1
      absh = 1 / rh;
    end
    absh = max(absh, hmin);
  else
    absh = min(hmax, max(hmin, htry));
  end
  f(:,1) = f0;

  % Initialize the output function.
  if haveoutfun
    feval(outfun,[t tfinal],y(outputs),'init');
  end

  % THE MAIN LOOP

  done = false;
  while ~done

    % By default, hmin is a small number such that t+hmin is only slightly
    % different than t. It might be 0 if t is 0.
    hmin = 16 * eps * abs(t);
    absh = min(hmax, max(hmin, absh));   % couldn't limit absh until new hmin
    h = tdir * absh;

    % Stretch the step if within 10% of tfinal-t.
    if 1.1 * absh >= abs(tfinal - t)
      h = tfinal - t;
      absh = abs(h);
      done = true;
    end

    % LOOP FOR ADVANCING ONE STEP.
    nofailed = true;                  % no failed attempts
    while true
      hA = h * A;
      hB = h * B;
      f(:,2) = feval(odefile, t + hA(1), y + f * hB(:,1), args{:});
```

```
f(:,3) = feval(odefile, t + hA(2), y + f * hB(:,2), args{:});
f(:,4) = feval(odefile, t + hA(3), y + f * hB(:,3), args{:});
f(:,5) = feval(odefile, t + hA(4), y + f * hB(:,4), args{:});
f(:,6) = feval(odefile, t + hA(5), y + f * hB(:,5), args{:});
tnew = t + hA(6);
ynew = y + f * hB(:,6);
f(:,7) = feval(odefile, tnew, ynew, args{:});
nfevals = nfevals + 6;                 % stats

% Estimate the error.
if normcontrol
  normynew = norm(ynew);
  err = absh * (norm(f * E) / max(max(normy,normynew),threshold));
else
  err = absh * norm((f * E) ./ max(max(abs(y),abs(ynew)),threshold),inf);
end

% Accept the solution only if the weighted error is no more than the
% tolerance rtol. Estimate an h that will yield an error of rtol on
% the next step or the next try at taking this step, as the case may be,
% and use 0.8 of this value to avoid failures.
if err > rtol        % Failed step
  nfailed = nfailed + 1;   % stats
  if absh <= hmin
    msg = sprintf(['Failure at t=%e. Unable to meet integration ' ...
                   'tolerances without reducing the step size below ' ...
                   'the smallest value allowed (%e) at time
t.\n'],t,hmin);
    warning(msg);
    if haveoutfun
      feval(outfun,[],[],'done');
    end
    if printstats       % print cost statistics
      fprintf('%g successful steps\n', nsteps);
      fprintf('%g failed attempts\n', nfailed);
      fprintf('%g function evaluations\n', nfevals);
      fprintf('%g partial derivatives\n', npds);
      fprintf('%g LU decompositions\n', ndecomps);
      fprintf('%g solutions of linear systems\n', nsolves);
    end
    if nargout > 0
      tout = tout(1:nout);
      yout = yout(1:nout,:);
      if haveeventfun
```

```
        o3 = teout;
        o4 = yeout;
        o5 = ieout;
        o6 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
      else
        o3 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
      end
    end
    return;
  end

    if nofailed
      nofailed = false;
      absh = max(hmin, absh * max(0.1, 0.8 * (rtol/err) ^ pow));
    else
      absh = max(hmin, 0.5 * absh);
    end
    h = tdir * absh;
    done = false;

  else                          % Successful step
    break;

  end
end
nsteps = nsteps + 1;    % stats

if haveeventfun
  [te,ye,ie,valt,stop] = ...
      odezero('ntrp45',odefile,valt,t,y,tnew,ynew,t0,varargin,h,f);
  nte = length(te);
  if nte > 0
    if nargout > 2
      teout = [teout; te];
      yeout = [yeout; ye.'];
      ieout = [ieout; ie];
    end
    if stop                     % stop on a terminal event
      tnew = te(nte);
      ynew = ye(:,nte);
      done = true;
    end
  end
end
```

```
if nargout > 0
  oldnout = nout;
  if outflag == 3        % computed points, with refinement
    nout = nout + refine;
    if nout > length(tout)
      tout = [tout; zeros(chunk,1)];    % requires chunk >= refine
      yout = [yout; zeros(chunk,neq)];
    end
    i = oldnout+1:nout-1;
    tout(i) = t + (tnew-t) * S;
    yout(i,:) = ntrp45(tout(i),t,y,[],[],h,f).';
    tout(nout) = tnew;
    yout(nout,:) = ynew.';
  elseif outflag == 2      % computed points, no refinement
    nout = nout + 1;
    if nout > length(tout)
      tout = [tout; zeros(chunk,1)];
      yout = [yout; zeros(chunk,neq)];
    end
    tout(nout) = tnew;
    yout(nout,:) = ynew.';
  elseif outflag == 1      % output only at tspan points
    while next <= ntspan
      if tdir * (tnew - tspan(next)) < 0
        if haveeventfun & done
          nout = nout + 1;
          tout(nout) = tnew;
          yout(nout,:) = ynew.';
        end
        break;
      elseif tnew == tspan(next)
        nout = nout + 1;
        tout(nout) = tnew;
        yout(nout,:) = ynew.';
        next = next + 1;
        break;
      end
      nout = nout + 1;       % tout and yout are already allocated
      tout(nout) = tspan(next);
      yout(nout,:) = ntrp45(tspan(next),t,y,[],[],h,f).';
      next = next + 1;
    end
  end
end
```

```
    if haveoutfun
      i = oldnout+1:nout;
      if ~isempty(i) & (feval(outfun,tout(i),yout(i,outputs).') == 1)
        tout = tout(1:nout);
        yout = yout(1:nout,:);
        if haveeventfun
          c3 = teout;
          c4 = yeout;
          c5 = ieout;
          c6 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
        else
          c3 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
        end
        return;
      end
    end
  elseif haveoutfun
    if outflag == 3          % computed points, with refinement
      tinterp = t + (tnew-t) * S;
      yinterp = ntrp45(tinterp,t,y,[],[],h,f);
      if feval(outfun,[tinterp; tnew],[yinterp(outputs,:), ynew(outputs)]) == 1
        return;
      end
    elseif outflag == 2
      if feval(outfun,tnew,ynew(outputs)) == 1
        return;
      end
    elseif outflag == 1       % output only at tspan points
      ninterp = 0;
      while next <= ntspan
        if tdir * (tnew - tspan(next)) < 0
          if haveeventfun & done
            ninterp = ninterp + 1;
            tinterp(ninterp,1) = tnew;
            yinterp(:,ninterp) = ynew;
          end
          break;
        elseif tnew == tspan(next)
          ninterp = ninterp + 1;
          tinterp(ninterp,1) = tnew;
          yinterp(:,ninterp) = ynew;
          next = next + 1;
          break;
```

```
        end
        ninterp = ninterp + 1;
        tinterp(ninterp,1) = tspan(next);
        yinterp(:,ninterp) = ntrp45(tspan(next),t,y,[],[],h,f);
        next = next + 1;
      end
      if ninterp > 0
        if feval(outfun,tinterp(1:ninterp),yinterp(outputs,1:ninterp)) == 1
          return;
        end
      end
    end
  end

  % If there were no failures compute a new h.
  if nofailed
    % Note that absh may shrink by 0.8, and that err may be 0.
    temp = 1.25 * (err/rtol) ^ pow;
    if temp > 0.2
      absh = absh / temp;
    else
      absh = 5.0 * absh;
    end
  end

  % Advance the integration one step.
  t = tnew;
  y = ynew;
  if normcontrol
    normy = normynew;
  end
  f(:,1) = f(:,7);          % Already evaluated odefile(tnew,ynew)
end

if haveoutfun
  feval(outfun,[],[],'done');
end

if printstats                   % print cost statistics
  fprintf('%g successful steps\n', nsteps);
  fprintf('%g failed attempts\n', nfailed);
  fprintf('%g function evaluations\n', nfevals);
  fprintf('%g partial derivatives\n', npds);
  fprintf('%g LU decompositions\n', ndecomps);
```

```
  fprintf('%g solutions of linear systems\n', nsolves);
end

if nargout > 0
  tout = tout(1:nout);
  yout = yout(1:nout,:);
  if haveeventfun
    o3 = teout;
    o4 = yeout;
    o5 = ieout;
    o6 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
  else
    o3 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
  end
end
```

64. function [tout,yout,o3,o4,o5,o6] = ode15s(odefile,tspan,y0,options,varargin)

函数 ODE15S 为解决刚性常微分方程的初值问题的通用程序。

【例子】

[T,Y] = ODE15S('F',TSPAN,Y0)

输入参数：

TSPAN 为积分计算区间矢量，如 TSPAN = [T0 TFINAL]；

Y0 为常微分组的初始值矢量；

字符串参数 F 为定义一个常微分方程组的调用函数。

输出参数：

T 积分时间序列；

Y 为在积分时间序列的常微分组的解矩阵。

使用示例：

1）先用 M 文件定义一个常微分方程组的调用函数 stiff1(t,x)：

```
function xdot=stiff1(t,x);
xdot(1)=x(1)*(-1+x(2))+.99*x(2);
xdot(2)=100*(x(1)-x(2)-x(1)*x(2));
xdot=xdot';
```

将此函数存为 M 文件。

2）继用下述步骤来调用函数 ode15s，并将结果在屏幕上显示：

```
>t0=0;
>tf=20;
>x0=[0  .25]
>[t,x]=ode15s('stiff1',[t0 tf],x0');
>plot(t,x)
```

有关详细使用细节，请参见"The MATLAB ODE Suite"，L. F. Shampine and M. W. Reichelt，SIAM Journal on Scientific Computing，18-1，1997.

【程序】

```
true = logical(1);
```

```
false = ~true.

nsteps = 0;                % stats
nfailed = 0;               % stats
nfevals = 0;               % stats
npds = 0;                  % stats
ndecomps = 0;              % stats
nsolves = 0;               % stats

if nargin == 0
  error('Not enough input arguments. See ODE15S.');
elseif ~isstr(odefile) & ~isa(odefile, 'inline')
  error('First argument must be a single--quoted string. See ODE15S.');
end
if nargin == 1
  tspan = []; y0 = []; options = [];
elseif nargin == 2
  y0 = []; options = [];
elseif nargin == 3
  options = [];
elseif ~isempty(options) & ~isa(options,'struct')
  if (length(tspan) == 1) & (length(y0) == 1) & (min(size(options)) == 1)
    tspan = [tspan; y0];
    y0 = options;
    options = [];
    varargin = {};
    msg = sprintf('Use ode15s("%s",tspan,y0,...) instead.',odefile);
    warning(['Obsolete syntax. ' msg]);
  else
    error('Correct syntax is ode15s("odefile",tspan,y0,options).');
  end
end
if isstr(odefile) & (exist(odefile) == 4) % a SIMULINK model
  warning (['Future versions of this M-file will not support '...
          'SIMULINK models. Use the SIM command directly.']);

  options=simset('Solver','ode15s',...
                 'InitialState', y0,...
                 'RelTol',odeget(options,'RelTol'),...
                 'AbsTol',odeget(options,'AbsTol'),...
                 'Refine',odeget(options,'Refine'),...
                 'MaxStep',odeget(options,'MaxStep'),...
                 'InitialStep',odeget(options,'InitialStep'),...
                 'MaxOrder',odeget(options,'MaxOrder'));
```

```
  % The code before and after the call to sim copies the ToWorkspace
  %      variables to the caller workspace.
  % Note: we must use unique variables names at this point to be
  %      distinct from ToWorkspace variable names.
  odefile_cp2wksp = odefile;
  [tout_cp2wksp,yout_cp2wksp,o3_cp2wksp] = ...
                sim(odefile,tspan,options,varargin{:});

  % Copy ToWorkspace variables to the caller workspace.
  names_cp2wksp = find_system(odefile_cp2wksp,'LookUnderMasks','on', ...
                                           'BlockType','ToWorkspace');
  for i_cp2wksp = 1:length(names_cp2wksp)
    varname_cp2wksp = get_param(names_cp2wksp{i_cp2wksp}, ...
                               'VariableName');
    assignin('caller', varname_cp2wksp, eval(varname_cp2wksp));
  end
  % Now safe to use tout, yout, and o3
  tout = tout_cp2wksp; yout = yout_cp2wksp; o3 = o3_cp2wksp;

  return;
end

% Get default tspan and y0 from odefile if none are specified.
if isempty(tspan) | isempty(y0)
  if (nargout(odefile) < 3) & (nargout(odefile) ~= -1)
    msg = sprintf('Use ode15s("%s",tspan,y0,...) instead.',odefile);
    error(['No default parameters in ' upper(odefile) '. ' msg]);
  end
  [def_tspan,def_y0,def_options] = feval(odefile,[],[],'init',varargin{:});
  if isempty(tspan)
    tspan = def_tspan;
  end
  if isempty(y0)
    y0 = def_y0;
  end
  if isempty(options)
    options = def_options;
  else
    options = odeset(def_options,options);
  end
end

% Test that tspan is internally consistent.
tspan = tspan(:);
```

```
ntspan = length(tspan);
if ntspan == 1
  t0 = 0;
  next = 1;
else
  t0 = tspan(1);
  next = 2;
end
tfinal = tspan(ntspan);
if t0 == tfinal
  error('The last entry in tspan must be different from the first entry.');
end
tdir = sign(tfinal - t0);
if any(tdir * (tspan(2:ntspan)-tspan(1:ntspan-1)) <= 0)
  error('The entries in tspan must strictly increase or decrease.');
end

t = t0;
y = y0(:);
neq = length(y);

% Get options, and set defaults.
rtol = odeget(options,'RelTol',1e-3);
if (length(rtol) ~= 1) | (rtol <= 0)
  error('RelTol must be a positive scalar.');
end
if rtol < 100 * eps
  rtol = 100 * eps;
  warning(['RelTol has been increased to ' num2str(rtol) '.']);
end

atol = odeget(options,'AbsTol',1e-6);
if any(atol <= 0)
  error('AbsTol must be positive.');
end

normcontrol = strcmp(odeget(options,'NormControl','off'),'on');
if normcontrol
  if length(atol) ~= 1
    error('Solving with NormControl "on" requires a scalar AbsTol.');
  end
  normy = norm(y);
else
  if (length(atol) ~= 1) & (length(atol) ~= neq)
```

```
      error(sprintf (['Solving %s requires a scalar AbsTol, ' ...
                     'or a vector AbsTol of length %d'],upper(odefile),neq));
  end
  atol = atol(:);
end
threshold = atol / rtol;

% By default, hmax is 1/10 of the interval.
hmax = min(abs(tfinal-t), abs(odeget(options,'MaxStep',0.1 * (tfinal-t))));
if hmax <= 0
  error('Option "MaxStep" must be greater than zero.');
end
htry = abs(odeget(options,'InitialStep'));
if htry <= 0
    error('Option "InitialStep" must be greater than zero.');
end

haveeventfun = strcmp(odeget(options,'Events','off'),'on');
if haveeventfun
  valt = feval(odefile,t,y,'events',varargin{:});
  teout = [];
  yeout = [];
  ieout = [];
end

if nargout > 0
   outfun = odeget(options,'OutputFcn');
else
  outfun = odeget(options,'OutputFcn','odeplot');
end
if isempty(outfun)
  haveoutfun = false;
else
  haveoutfun = true;
  outputs = odeget(options,'OutputSel',1:neq);
end
refine = odeget(options,'Refine',1);
printstats = strcmp(odeget(options,'Stats','off'),'on');

Janalytic = strcmp(odeget(options,'Jacobian','off'),'on');
Jconstant = strcmp(odeget(options,'Jconstant   ','off'),'on');
Vectorized = strcmp(odeget(options,'Vectorized   ','off'),'on');
Jpattern = strcmp(odeget(options,'Jpattern','off'),'on');
if Jpattern
```

```
  Js = feval(odefile,[],[],'jpattern',varargin{:});
else
  Js=[];
end

mass = strcmp(odeget(options,'Mass','off'),'on');
if mass
  Mt = feval(odefile,t,[],'mass',varargin{:});
  Mconstant = strcmp(odeget(options,'MassConstant','off'),'on');
else
  Mt = sparse((1:neq)',(1:neq)',1,neq,neq);% sparse Identity matrix
  Mconstant=true;
end
if Mconstant
  Mtnew = Mt;
end

maxk = odeget(options,'MaxOrder',5);
bdf = strcmp(odeget(options,'BDF','off'),'on');

% Set the output flag.
if ntspan > 2
  outflag = 1;                          % output only at tspan points
elseif refine <= 1
  outflag = 2;                          % computed points, no refinement
else
  outflag = 3;                          % computed points, with refinement
  S = (1:refine-1)' / refine;
end

% Allocate memory if we're generating output.
if nargout > 0
  if ntspan > 2                         % output only at tspan points
    tout = zeros(ntspan,1);
    yout = zeros(ntspan,neq);
  else                                  % alloc in chunks
    chunk = max(ceil(128 / neq),refine);
    tout = zeros(chunk,1);
    yout = zeros(chunk,neq);
  end
  nout = 1;
  tout(nout) = t;
  yout(nout,:) = y.';
end
```

```
% Initialize method parameters.
G = [1; 3/2; 11/6; 25/12; 137/60];
if bdf
  alpha = [0; 0; 0; 0; 0];
else
  alpha = [-37/200; -1/9; -0.0823; -0.0415; 0];
end
invGa = 1 ./ (G .* (1 - alpha));
erconst = alpha .* G + (1 ./ (2:6)');
difU = [ -1,  -2,  -3,  -4,  -5;              % difU is its own inverse!
          0,   1,   3,   6,  10;
          0,   0,  -1,  -4, -10;
          0,   0,   0,   1,   5;
          0,   0,   0,   0,  -1 ];
maxK = 1 maxk;
[kJ,kI] = meshgrid(maxK,maxK);
difU = difU(maxK,maxK);
maxit = 4;

% Maintain backward compatibility with ODE files that don't accept flags.
if isempty(varargin)
  args = {};
else                                          % There are extra arguments, so
  args = [{[]} varargin];                     % stick an empty flag on the front
end                                           % before sending them to odefile.

f0 = feval(odefile,t,y,args{:});
nfevals = nfevals + 1;                        % stats
[m,n] = size(f0);
if n > 1
  error([upper(odefile) ' must return a column vector.'])
elseif m ~= neq
  msg = sprintf('an initial condition vector of length %d.',m);
  error(['Solving ' upper(odefile) ' requires ' msg]);
end

if Janalytic
  dfdy = feval(odefile,t,y,'jacobian',varargin{:});
else
  jthresh = atol + zeros(neq,1);
  [dfdy,fac,g,nF] = ...
     numjac(odefile,t,y,f0,jthresh,[],vectorized,Js,[],args{:});
  nfevals = nfevals + nF;                     % stats
```

```
end
npds = npds + 1;                               % stats
JMcurrent = true;

hmin = 16 * eps * abs(t);
if isempty(htry)
  % Compute an initial step size h using y'(t).
  absh = min(hmax, abs(tspan(next)-t));
  if normcontrol
    wt = max(normy,threshold);
    if mass
      [L,U] = lu(Mt);
      ndecomps = ndecomps + 1;                 % stats
      F0 = U \ (L \ f0);
      nsolves = nsolves + 1;                   % stats
      rh = (norm(F0) / wt) / (0.8 * sqrt(rtol));
    else
      rh = (norm(f0) / wt) / (0.8 * sqrt(rtol));
    end
  else
    wt = max(abs(y),threshold);
    if mass
      [L,U] = lu(Mt);
      ndecomps = ndecomps + 1;                 % stats
      F0 = U \ (L \ f0);
      nsolves = nsolves + 1;                   % stats
      rh = norm(F0 ./ wt,inf) / (0.8 * sqrt(rtol));
    else
      rh = norm(f0 ./ wt,inf) / (0.8 * sqrt(rtol));
    end
  end
  if absh * rh > 1
    absh = 1 / rh;
  end
  absh = max(absh, hmin);

  % The error of BDF1 is 0.5 * h ^ 2 * y''(t), so we can determine the optimal h.
  h = tdir * absh;
  tdel = (t + tdir * min(sqrt(eps) * max(abs(t),abs(t+h)),absh)) - t;
  f1 = feval(odefile,t+tdel,y,args{:});
  nfevals = nfevals + 1;                       % stats
  dfdt = (f1-f0) ./ tdel;
  absh = min(hmax, abs(tspan(next)-t));
  if normcontrol
```

```
      if mass
        rh = 1.25 * sqrt(0.5 * (norm(U \ (L \ (dfdt + dfdy * F0))) / wt) / rtol);
      else
        rh = 1.25 * sqrt(0.5 * (norm(dfdt + dfdy * f0) / wt) / rtol);
      end
    else
      if mass
        rh = 1.25 * sqrt(0.5 * norm((U \ (L \ (dfdt + dfdy * F0))) ./ wt,inf) / rtol);
      else
        rh = 1.25 * sqrt(0.5 * norm((dfdt + dfdy * f0) ./ wt,inf) / rtol);
      end
    end
    if absh * rh > 1
      absh = 1 / rh;
    end
    absh = max(absh, hmin);
  else
    absh = min(hmax, max(hmin, htry));
  end

  % Initialize.
  k = 1;                                  % start at order 1 with BDF1
  K = 1;                                  % K = 1:k
  klast = k;
  abshlast = absh;

  h = tdir * absh;
  dif = zeros(neq,maxk+2);
  if mass
    dif(:,1) = h * F0;
  else
    dif(:,1) = h * f0;
  end

  hinvGak = h * invGa(k);
  nconhk = 0;                             % steps taken with current h and k
  [L,U] = lu(Mt-hinvGak * dfdy);
  ndecomps = ndecomps + 1;                % stats
  havrate = false;

  % Initialize the output function.
  if haveoutfun
    feval(outfun,[t tfinal],y(outputs),'init');
  end
```

```
% THE MAIN LOOP

done = false;
while ~done
  % By default, hmin is a small number such that t+hmin is only slightly
  % different than t. It might be 0 if t is 0.
  hmin = 16 * eps * abs(t);
  absh = min(hmax, max(hmin, absh));          % couldn't limit absh until new hmin
  h = tdir * absh;

  % Stretch the step if within 10% of tfinal-t.
  if 1.1 * absh >= abs(tfinal-t)
    h = tfinal-t;
    absh = abs(h);
    done = true;
  end

  if (absh ~= abshlast) | (k ~= klast)
    difRU = cumprod((kI - 1 - kJ * (absh/abshlast)) ./ kI) * difU;
    dif(:,K) = dif(:,K) * difRU(K,K);

    hinvGak = h * invGa(k);
    nconhk = 0;
    [L,U] = lu(Mt-hinvGak * dfdy);
    ndecomps = ndecomps + 1;                  % stats
    havrate = false;
  end

  % LOOP FOR ADVANCING ONE STEP.
  nofailed = true;                            % no failed attempts
  while true

    gotynew = false;                          % is ynew evaluated yet?
    while ~gotynew

      % Compute the constant terms in the equation for ynew.
      psi = dif(:,K) * (G(K) * invGa(k));

      % Predict a solution at t+h.
      tnew = t + h;
      pred = y + sum(dif(:,K),2);
      ynew = pred;
```

```
% The difference, difkp1, between pred and the final accepted
% ynew is equal to the backward difference of ynew of order
% k+1. Initialize to zero for the iteration to compute ynew.
difkp1 = zeros(neq,1);
if normcontrol
  normynew = norm(ynew);
  invwt = 1 / max(max(norm(y),normynew),threshold);
  minnrm = 100 * eps * (normynew * invwt);
else
  invwt = 1 ./ max(max(abs(y),abs(ynew)),threshold);
  minnrm = 100 * eps * norm(ynew .* invwt,inf);
end

% Mtnew is required in the RHS function evaluation.
if ~Mconstant
  Mtnew = feval(odefile,t,[],'mass',varargin{:});
end

% Iterate with simplified Newton method.
tooslow = false;
for iter = 1:maxit
  del = U \ (L \ (hinvGak * feval(odefile,tnew,ynew,args{:}) - ...
      Mtnew * (psi+difkp1)));
  if normcontrol
  newnrm = norm(del) * invwt;
else
  newnrm = norm(del .* invwt,inf);
end
difkp1 = difkp1 + del;
ynew = pred + difkp1;

if newnrm <= minnrm
  gotynew = true;
  break;
elseif iter == 1
  if havrate
    errit = newnrm * rate / (1-rate);
    if errit <= 0.05 * rtol                % More stringent when using old rate.
      gotynew = true;
      break;
    end
  else
    rate = 0;
  end
```

```
    elseif newnrm > 0.9 * oldnrm
      tooslow = true;
      break;
    else
      rate = max(0.9 * rate, newnrm / oldnrm);
      havrate = true;
      errit = newnrm * rate / (1 - rate);
      if errit <= 0.5 * rtol
        gotynew = true;
        break;
      elseif iter == maxit
        tooslow = true;
        break;
      elseif 0.5 * rtol < errit * rate ^ (maxit-iter)
        tooslow = true;
        break;
      end
    end

    oldnrm = newnrm;
  end                                          % end of Newton loop
  nfevals == nfevals + iter;                      % stats
  nsolves == nsolves + iter;                      % stats

  if tooslow
    nfailed = nfailed + 1;                        % stats
    % Speed up the iteration by forming J(t,y) or reducing h.
      if ~JMcurrent
        if ~Jconstant
          if Janalytic
            dfdy = feval(odefile,t,y,'jacobian',varargin{:});
          else
            f0 = feval(odefile,t,y,args{:});
            [dfdy,fac,g,nF] = ...
                numjac(odefile,t,y,f0,jthresh,fac,vectorized,Js,g,args{:});
            nfevals = nfevals + nF + 1; % stats
          end
            npds = npds + 1;                      % stats
          end
          JMcurrent = true;
        elseif absh <= hmin
          msg = sprintf(['Failure at t= %e. Unable to meet integration ' ...
                        'tolerances without reducing the step size below ' ...
                        'the smallest value allowed (%e) at time t.\n'], ...
```

```
                        t,hmin);
        warning(msg);
        if haveoutfun
          feval(outfun,[],[],'done');
        end
        if printstats                          % print cost statistics
          fprintf('%g successful steps\n', nsteps);
          fprintf('%g failed attempts\n', nfailed);
          fprintf('%g function evaluations\n', nfevals);
          fprintf('%g partial derivatives\n', npds);
          fprintf('%g LU decompositions\n', ndecomps);
          fprintf('%g solutions of linear systems\n', nsolves);
        end
        if nargout > 0
          tout = tout(1:nout);
          yout = yout(1:nout,:);
          f haveeventfun
            o3 = teout;
            o4 = yeout;
            o5 = ieout;
            o6 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
          else
            o3 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
          end
        end
        return;
      else
        abshlast = absh;
        absh = max(0.3 * absh, hmin);
        h = tdir * absh;
        done = false;

        difRU = cumprod((kI-1-kJ * (absh/abshlast)) ./ kI) * difU;
        dif(:,K) = dif(:,K) * difRU(K,K);

        hinvGak = h * invGa(k);
        nconhk = 0;
      end
      [L,U] = lu(Mt-hinvGak * dfdy);
      ndecomps = ndecomps + 1;               % stats
      havrate = false;
    end
end         % end of while loop for getting ynew
```

```
  % difkp1 is now the backward difference of ynew of order k+1.
  if normcontrol
    err = (norm(difkp1) * invwt) * erconst(k);
  else
    err = norm(difkp1 .* invwt,inf) * erconst(k);
  end

  if err > rtol                                    % Failed step
    nfailed = nfailed + 1;                         % stats
    if absh <= hmin
      msg = sprintf (['Failure at t=%e. Unable to meet integration ' ...
                      'tolerances without reducing the step size below ' ...
                      'the smallest value allowed (%e) at time
t.\n'],t,hmin);
      warning(msg);
      if haveoutfun
        feval(outfun,[],[],'done');
      end
      if printstats                                % print cost statistics
        fprintf('%g successful steps\n', nsteps);
        fprintf('%g failed attempts\n', nfailed);
        fprintf('%g function evaluations\n', nfevals);
        fprintf('%g partial derivatives\n', npds);
        fprintf('%g LU decompositions\n', ndecomps);
        fprintf('%g solutions of linear systems\n', nsolves);
      end
      if nargout > 0
        tout = tout(1:nout);
        yout = yout(1:nout,:);
        if haveeventfun
          o3 = teout;
          o4 = yeout;
          o5 = ieout;
          o6 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
        else
          o3 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
        end
      end
      return;
    end

    abshlast = absh;
    if nofailed
      nofailed = false;
```

```
      hopt = absh * max(0.1, 0.833 * (rtol/err) ^ (1/(k-1))); % 1/1.2
      if k > 1
        if normcontrol
          errkm1 = (norm(dif(:,k) + difkp1) * invwt) * erconst(k-1);
        else
          errkm1 = norm((dif(:,k) + difkp1) .* invwt,inf) * erconst(k-1);
        end
        hkm1 = absh * max(0.1, 0.769 * (rtol/errkm1) ^ (1/k)); % 1/1.3
        if hkm1 > hopt
          hopt = min(absh,hkm1);          % don't allow step size increase
          k = k - 1;
          K = 1:k;
        end
      end
      absh = max(hmin, hopt);
    else
      absh = max(hmin, 0.5 * absh);
    end
    h = tdir * absh;
    if absh < abshlast
      done = false;
    end

    difRU = cumprod((kI-1-kJ * (absh/abshlast)) ./ kI) * difU;
    dif(:,K) = dif(:,K) * difRU(K,K);

    hinvGak = h * invGa(k);
    nconhk = 0;
    [L,U] = lu(Mt - hinvGak * dfdy);
    ndecomps = ndecomps + 1;              % stats
    havrate = false;

  else                                    % Successful step
    break;

  end
end
nsteps = nsteps + 1;                      % stats

dif(:,k+2) = difkp1-dif(:,k+1);
dif(:,k+1) = difkp1;
for j = k:-1:1
  dif(:,j) = dif(:,j) + dif(:,j+1);
end
```

```
tstep = tnew;
ystep = ynew;
if haveeventfun
  [te,ye,ie,valt,stop] = ...
    odezero('ntrp15s',odefile,valt,t,y,tnew,ynew,t0,varargin,h,dif,k);
  nte = length(te);
  if nte > 0
    if nargout > 2
      teout = [teout; te];
      yeout = [yeout; ye.'];
      ieout = [ieout; ie];
    end
    if stop                           % stop on a terminal event
      tnew = te(nte);
      ynew = ye(:,nte);
      done = true;
    end
  end
end

if nargout > 0
  oldnout = nout;
  if outflag == 2                     % computed points, no refinement
    nout = nout + 1;
    if nout > length(tout)
      tout = [tout; zeros(chunk,1)];
      yout = [yout; zeros(chunk,neq)];
    end
    tout(nout) = tnew;
    yout(nout,:) = ynew.';
  elseif outflag == 3                 % computed points, with refinement
    nout = nout + refine;
    if nout > length(tout)
      tout = [tout; zeros(chunk,1)]; % requires chunk >= refine
      yout = [yout; zeros(chunk,neq)];
    end
    i = oldnout+1:nout-1;
    tout(i) = t + (tnew-t)*S;
    yout(i,:) = ntrp15s(tout(i),[],[],tstep,ystep,h,dif,k).';
    tout(nout) = tnew;
    yout(nout,:) = ynew.';
  elseif outflag == 1                 % output only at tspan points
    while next <= ntspan
```

```
        if tdir * (tnew-tspan(next)) < 0
          if haveeventfun & done
            nout = nout + 1;
            tout(nout) = tnew;
            yout(nout,:) = ynew.';
          end
          break;
        elseif tnew == tspan(next)
          nout = nout + 1;
          tout(nout) = tnew;
          yout(nout,:) = ynew.';
          next = next + 1;
          break;
        end
        nout = nout + 1;                          % tout and yout are already allocated
        tout(nout) = tspan(next);
        yout(nout,:) =
ntrp15s(tspan(next),[],[],tstep,ystep,h,dif,k).';
        next = next + 1;
      end
    end

    if haveoutfun
      i = oldnout+1:nout;
      if ~isempty(i) & (feval(outfun,tout(i),yout(i,outputs).') == 1)
        tout = tout(1:nout);
        yout = yout(1:nout,:);
        if haveeventfun
          o3 = teout;
          o4 = yeout;
          o5 = ieout;
          o6 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
        else
          o3 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
        end
        return;
      end
    end

  elseif haveoutfun
    if outflag == 2
      if feval(outfun,tnew,ynew(outputs)) == 1
        return;
      end
```

```
    elseif outflag == 3                          % computed points, with refinement
      tinterp = t + (tnew-t) * S;
      yinterp = ntrp15s(tinterp,[],[],tstep,ystep,h,dif,k);
      if feval(outfun,[tinterp; tnew],[yinterp(outputs,:), ynew(outputs)]) == 1
        return;
      end
    elseif outflag == 1                          % output only at tspan points
      ninterp = 0;
      while next <= ntspan
        if tdir * (tnew-tspan(next)) < 0
          if haveeventfun & done
            ninterp = ninterp + 1;
            tinterp(ninterp,1) = tnew;
            yinterp(:,ninterp) = ynew;
          end
          break;
        elseif tnew == tspan(next)
          ninterp = ninterp + 1;
          tinterp(ninterp,1) = tnew;
          yinterp(:,ninterp) = ynew;
          next = next + 1;
          break;
        end
        ninterp = ninterp + 1;
        tinterp(ninterp,1) = tspan(next);
        yinterp(:,ninterp) =
ntrp15s(tspan(next),[],[],tstep,ystep,h,dif,k);
        next = next + 1;
      end
      if ninterp > 0
        if feval(outfun,tinterp(1:ninterp),yinterp(outputs,1:ninterp)) == 1
          return;
        end
      end
    end
  end

  klast = k;
  abshlast = absh;
  nconhk = min(nconhk+1,maxk+2);
  if nconhk >= k + 2
    temp = 1.2 * (err/rtol)^(1/(k+1));
    if temp > 0.1
      hopt = absh / temp;
```

```
else
  hopt == 10 * absh;
end
kopt = k;
if k > 1
  if normcontrol
    errkm1 == (norm(dif(:,k)) * invwt) * erconst(k - 1);
  else
    errkm1 = norm(dif(:,k) . * invwt,inf) * erconst(k   1);
  end
  temp = 1.3 * (errkm1/rtol)^ (1/k);
  if temp > 0.1
    hkm1 = absh / temp;
  else
    hkm1 = 10 * absh;
  end
  if hkm1 > hopt
    hopt = hkm1;
    kopt = k - 1;
  end
end
if k < maxk
  if normcontrol
    errkp1 = (norm(dif(:,k+2)) * invwt) * erconst(k + 1);
  else
    errkp1 — norm(dif(:,k+2) . * invwt,inf) * erconst(k-1);
  end
  temp — 1.4 * (errkp1/rtol)^ (1/(k+2));
  if temp > 0.1
    hkp1 = absh / temp;
  else
    hkp1 = 10 * absh;
  end
  if hkp1 > hopt
    hopt = hkp1;
    kopt = k + 1;
  end
end
if hopt > absh
  absh = hopt;
  if k ~= kopt
    k = kopt;
    K = 1:k;
  end
```

```
    end
  end

  % Advance the integration one step.
  t = tnew;
  y = ynew;
  JMcurrent = Jconstant & Mconstant;
  if ~Mconstant
    Mt = Mtnew;
  end

end

if haveoutfun
  feval(outfun,[],[],'done');
end
if printstats                              % print cost statistics
  fprintf('%g successful steps\n', nsteps);
  fprintf('%g failed attempts\n', nfailed);
  fprintf('%g function evaluations\n', nfevals);
  fprintf('%g partial derivatives\n', npds);
  fprintf('%g LU decompositions\n', ndecomps);
  fprintf('%g solutions of linear systems\n', nsolves);
end
if nargout > 0
  tout = tout(1:nout);
  yout = yout(1:nout,:);
  if haveeventfun
    o3 = teout;
    o4 = yeout;
    o5 = ieout;
    o6 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
  else
    o3 = [nsteps; nfailed; nfevals; npds; ndecomps; nsolves];
  end
end
```

65. function a = appcoef(c,l,arg3,arg4,arg5)

函数 APPCOEF 用于计算一维信号的逼近系数。

【例子】

(1) 函数　A = APPCOEF(C,L,'wname',N)

用于计算一维信号的小波分解水平 N 的逼近系数

其中参数：

structure [C,L] (参见 WAVEDEC);

'wname' 为一字符串变量，用以表示所用小波名；

水平 N 必须为一整数 0 <= N <=(L)的长度－2.

(2) 函数　A = APPCOEF(C,L,'wname')

用于计算一维信号的小波分解水平 N=(L) 的长度－2的逼近系数其他参数意义同前。

(3) 函数　A = APPCOEF(C,L,Lo_R,Hi_R) or

　　　　　A = APPCOEF(C,L,Lo_R,Hi_R,N),

其中参数 Lo_R 是重构低通滤波器及 Hi_R 是重构低通滤波器。其 他参数意义同前。

参见 DETCOEF，WAVEDEC.

【程序】

```
% Check arguments.
if errargn('appcoef',nargin,[3:5],nargout,[0:1]), error('*'), end
rmax = length(l);
nmax = rmax-2;
if isstr(arg3)
        [LoF_R,HiF_R] = wfilters(arg3,'r');
        if nargin == 4, n = arg4; else, n = nmax; end
else
        LoF_R = arg3; HiF_R = arg4;
        if nargin == 5, n = arg5; else, n = nmax; end
end
if (n < 0) | (n > nmax) | (n ~= fix(n))
        errargt('appcoef','invalid level value','msg'); error('*');
end

% Initialization.
a = c(1:l(1));

% Iterated reconstruction.
imax = rmax+1;
for p = nmax:-1:n+1
        d = detcoef(c,l,p);                    % extract detail
        a = idwt(a,d,LoF_R,HiF_R,l(imax-p));
end
```

66. function F = coifwavf(wname)

函数 COIFWAVF 用于计算 Coiflets 小波滤波器.

【例子】

F = COIFWAVF (W)

计算尺度滤波器。(scaling filter)

其中字符串参数：

W = 'coifN';

N 的可能数值为:N = 1, 2, 3, 4 or 5。

参见 WAVEINFO.

【程序】

```
% Check arguments.
```

```
if errargn('coifwavf',nargin,[1],nargout,[0:1]), error('*'); end

lw = length(wname); ab = abs(wname);
ii = lw + 1;
while (47<ab(ii-1)) & (ab(ii-1)<58) , ii = ii-1; end
num = str2num(wname(ii:lw));

if        num==1
F = [...
  -0.05142972847100   0.23892972847100   0.60285945694200   0.27214054305800  ...
  -0.05142972847100  -0.01107027152900  ...
                ];

elseif num==2
F = [...
  0.01158759673900  -0.02932013798000  -0.04763959031000   0.27302104653500  ...
  0.57468239385700   0.29486719369600  -0.05408560709200  -0.04202648046100  ...
  0.01674441016300   0.00396788361300  -0.00128920335600  -0.00050950539900  ...
                ];

elseif num==3
F = [...
  -0.00268241867100   0.00350312670900   0.01658356047900  -0.04650776447900  ...
  -0.04322076356000   0.28650333527400   0.56128525687000   0.30298357177300  ...
  -0.05077014075500  -0.05819625076200   0.02443409432100   0.01122924096200  ...
  -0.00636960101100  -0.00182045891600   0.00079020510100   0.00032966517400  ...
  -0.00005019277500  -0.00002446573400
                ];

elseif num==4
F = [...
  0.00063096104600  -0.00115222485200  -0.00519452402600   0.01136245924400  ...
  0.01886723537800  -0.05746423442900  -0.03965264851700   0.29366739089500  ...
  0.55312645256200   0.30715732619800  -0.04711273886500  -0.06803812705100  ...
  0.02781364015300   0.01773583743800  -0.01075631851700  -0.00400101288600  ...
  0.00265266594600   0.00089559452900  -0.00041650057100  -0.00018382976900  ...
  0.00004408035400   0.00002208285700  -0.00000230494200  -0.00000126217500  ...
              ];

elseif num==5
F = [...
  -0.00014996380000   0.00025356120000   0.00154024570000  -0.00294111080000  ...
  -0.00716378190000   0.01655206640000   0.01991780430000  -0.06499726280000  ...
  -0.03680007360000   0.29809232350000   0.54750542940000   0.30970684900000  ...
```

```
 -0.04386605080000  -0.07465223890000  0.02919587950000  0.02311077700000  ...
 -0.01397368790000  -0.00648009000000  0.00478300140000  0.00172065470000  ...
 -0.00117582220000  -0.00045122700000  0.00021372980000  0.00009937760000  ...
 -0.00002923210000  -0.00001507200000  0.00000264080000  0.00000145930000  ...
  0.00000011840000  -0.00000006730000
            ];
else
    errargt('coifwavf','Invalid coiflet','msg');
    error(' * ');
end
```

67. function w = dbaux(N,sumw);

函数 DBAUX 用于计算 Daubechies 小波滤波器

【例子】

W = DBAUX(N,SUMW)

用于计算尺度为 N 的 Daubechies 尺度滤波器。scaling filter such that SUM(W) = SUMW.

N 的可能取值为：N = 1, 2, 3, …

注意：当 N 太大时可能会引起不稳定。

W = DBAUX(N) 就等于 W = DBAUX(N,1).

W = DBAUX(N,0) 也等于 W = DBAUX(N,1).

参见 DEWAVF，WFILTERS.

【程序】

```
% Check arguments.
if errargn('dbaux',nargin,[1:2],nargout,[0:1]), error(' * '); end

if nargin < 2 , sumw = 1; end
if sumw == 0 , sumw = 1; end

% Order n daub scaling filter computation
% using "Lagrange" filters.
lon = 2 * N - 1;
sup = [-N+1:N];
a = zeros(1,N);
for k = 1:N
    nok = sup(sup ~= k);
    a(k) = prod(0.5-nok)/prod(k-nok);
end
P = dyadup(a,0);
P = [wrev(P),1,P ]; % "Lagrange a trous" filter such that
                    % P = conv(wrev(w),w) * 2 if w
                    % denotes the order N daub scaling filter.
rP = roots(P);
[s,K] = sort(abs(rP)); rP = rP(K);
[s,K] = sort(abs(rP+1)); rL = rP(K(lon+2:2 * lon));
[s,K] = sort(abs(rL)); rL = rL(K);
```

```
% rL gives partial root location of w.
% w have N zcros located at   1.
w = real(poly([rL(abs(rL)<1);-ones(N,1)]));
w = sumw * (w/sum(w));
```

68. function F = coifwavf(wname)

(1) F = COIFWAVF(W)

计算尺度滤波器。(scaling filter)

其中字符串参数 W = 'coifN'.

N 的可能数值为:N = 1, 2, 3, 4 or 5.

参见 WAVEINFO.

(2) function F = dbwavf(wname)

函数 DBWAVF 用于计算 Daubechies 小波滤波器.

【例子】

F = DBWAVF(W)

计算 Daubechies 小波尺度滤波器。(scaling filter)

其中字符串参数

W = 'dbN'.

N 的可能取值为:

N = 1, 2, 3, ⋯ , 50.

参见 DBAUX, WAVEINFO, WFILTERS.

【程序】

```
% Check arguments.
if errargn('dbwavf',nargin,[1],nargout,[0:1]), error('*'); end

if lower(wname(1:2))=='ha'
        num = 1;
else
        lw = length(wname); ab = abs(wname);
        ii = lw+1;
        while (47<ab(ii-1)) & (ab(ii-1)<58) , ii = ii-1; end
        num = str2num(wname(ii:lw));
end
if      num==1
F = [...
  0.50000000000000   0.50000000000000   ...
                  ];

elseif num==2
F = [...
  0.34150635094622   0.59150635094587   0.15849364905378   -0.09150635094587   ...
                  ];
elseif num==3
F = [...
```

```
    0.23523360389270   0.57055845791731   0.32518250026371   -0.09546720778426   ...
    -0.06041610415535   0.02490874986589   ...
                ];
elseif num==4
F = [...
    0.16290171402562   0.50547285754565   0.44610006912319   0.01978751311791   ...
    -0.13225358368437   0.02180815023739   0.02325180053556   -0.00749349466513   ...
             ];

elseif num==5
F = [...
    0.11320949129173   0.42697177135271   0.51216347213016   0.09788348067375   ...
    -0.17132835769133   -0.02280056594205   0.05485132932108   0.00441340005433   ...
    -0.00889593505093   0.00235871396920   ...
           ];

elseif num==6
F = [...
    0.07887121600143   0.34975190703757   0.53113187994121   0.22291566146505   ...
    -0.15999329944587   -0.09175903203003   0.06894404648720   0.01946160485396   ...
    -0.02233187416548   0.00039162557603   0.00337803118151   0.00076176690258   ...
                ];

elseif num==7
F = [...
    0.05504971537285   0.28039564181304   0.51557424581833   0.33218624110566   ...
    -0.10175691123173   -0.15841750564054   0.05042323250485   0.05700172257986   ...
    -0.02689122629486   -0.01171997078235   0.00887489618962   0.00030375749776   ...
    -0.00127395235906   0.00025011342658   ...
                ];

elseif num==8
F = [...
    0.03847781105406   0.22123362357624   0.47774307521438   0.41390826621166   ...
    -0.01119286766665   -0.20082931639111   0.00033409704628   0.09103817842345   ...
    -0.01228195052300   -0.03117510332533   0.00988607964808   0.00618442240954   ...
    -0.00344385962813   -0.00027700227421   0.00047761485533   -0.00008306863060   ...
                ];

elseif num==9
F = [...
    0.02692517479416   0.17241715192471   0.42767453217028   0.46477285717278   ...
    0.09418477475112   -0.20737588089628   -0.06847677451090   0.10503417113714   ...
    0.02172633772990   -0.04782363205882   0.00017744640673   0.01581208292614   ...
```

```
  -0.00333981011324  -0.00302748028715  0.00130618364018  0.00016290733601  ...
  -0.00017816487955  0.00002782275679
               ];

elseif num==10
F = [...
  0.01885857879640  0.13306109139687  0.37278753574266  0.48681405536610  ...
  0.19881887088440  -0.17666810089647  -0.13855493935993  0.09006372426666  ...
  0.06580149355070  -0.05048328559801  -0.02082962404385  0.02348490704841  ...
  0.00255021848393  -0.00758950116768  0.00098666268244  0.00140884329496  ...
  -0.00048497391996  -0.00008235450295  0.00006617718320  -0.00000937920789  ...
               ];

elseif (11<=num) & (num<=50)
        F = dbaux(num);

else
        error(' * * * Invalid Daubechies wavelet number ! * * * ');
end
```

69. function [thr,sorh,keepapp,crit] = ddencmp(dorc,worwp,x)

函数 DDENCMP 用于计算去噪或压缩预置值。

【例子】

(1) [THR,SORH,KEEPAPP,CRIT] = DDENCMP(IN1,IN2,X)

返回可用于小波或小波包去噪或压缩的预置值.

输入参数：

IN1 可为 'den'(表示去噪 de noising)或为 'cmp'表示压缩(compression)；

IN2 可为 'wv' (表示小波)或为 'wp' (表示小波包)；

X 可为一维信号(矢量)或二维信号(矩阵).

输出参数：

THR 为阈值 (threshold)，SORH 表示软(soft) 或硬 (hard)方法置阈值；

KEEPAPP 允许用户保留逼近系数(approximation coefficients)；参数 CRIT 只用于小波包。

(2) 对于小波(只需三个输出参数)，即

[THR,SORH,KEEPAPP] = DDENCMP(IN1,'wv',X)

返回可用于小波去噪(如果 IN1 = 'den') 或压缩(如果 IN1 = 'cmp')的预置值，所得预置值可用于 WDENCMP 函数

(3) 对于小波包(需四个输出参数)，即

[THR,SORH,KEEPAPP,CRIT] = DDENCMP(IN1,'wp',X)

返回可用于小波去噪(如果 IN1 = 'den') 或压缩(如果 IN1 = 'cmp')的预置值，所得预置值可用于 WDENCMP 函数

参见 WDENCMP, WENTROPY, WPDENCMP.

【程序】

```
% Check arguments.
if length(worwp) ~= 2
  errargt('ddencmp','invalid argument value','msg'); error(' * ')
```

```
end
if worwp == 'wv'
  if errargn('ddencmp',nargin,[3],nargout,[0:3]), error('*'), end
elseif worwp == 'wp'
  if errargn('ddencmp',nargin,[3],nargout,[0:4]), error('*'), end
else
  errargt('ddencmp','invalid argument value','msg'); error('*')
end
if length(dorc) ~= 3
  errargt('ddencmp','invalid argument value','msg'); error('*')
end
if (dorc ~= 'den') & (dorc ~= 'cmp')
  errargt('ddencmp','invalid argument value','msg'); error('*')
end

% Set problem dimension.
dim = 1;
if min(size(x)) ~= 1, dim = 2; end

% Set keepapp default value.
keepapp = ..;

% Set sorh default value.
sorh = 'h';
if dorc == 'den'
  if worwp == 'wv', sorh = 's'; end
end

% Set threshold default value.
n = prod(size(x));

% nominal threshold.
if dorc == 'den'
  if worwp == 'wv'                    % wavelets case.
          thr = sqrt(2*log(n));
  else                                % wavelet packets case.
          thr = sqrt(2*log(n*log(n)/log(2)));
  end
else
  thr = 1;
end

% rescaled threshold.
if dim == 1
```

```
        [c,l] = wavedec(x,1,'db1');
        c = c(l(1)+1:length(c));
else
        [c,l] = wavedec2(x,1,'db1');
        c = c(prod(l(1,:))-1:length(c));
end

normaliz = median(abs(c));
% if normaliz=0 in compression, kill the lowest coefs.
if dorc == 'cmp' & normaliz == 0
        normaliz = 0.05 * max(abs(c));
end

if dorc == 'den'
        if worwp == 'wv'
        thr = thr * normaliz/0.6745;
    else
        crit = 'sure';
    end
else
        thr = thr * normaliz;
        if worwp == 'wp', crit = 'threshold'; end
end
```

70. function d = detcoef(c,l,n)

函数 DETCOEF 用于计算一维信号的细节系数(detail coefficients).

【例子】

(1) D = DETCOEF(C,L,N)

从小波分解结构[C,L]来计算水平 N 的信号细节系数

at level N from the wavelet decomposition structure

(参见 WAVEDEC).

水平 N 必须是一个整数,即 1 <= N <= (L)的长度-2.

(2) D = DETCOEF(C,L)

从小波分解结构[C,L]来计算水平 n = (L) 的长度-2 的信号细节系数。

参见 APPCOEF, WAVEDEC.

【程序】

```
% Check arguments.
if errargn('detcoef',nargin,[2:3],nargout,[0:1]), error(' * ');end
rmax = length(l);
nmax = rmax-2;
if nargin==2 , n = nmax; end
if (n < 1) | (r > nmax) | (n ~= fix(n))
        errargt('detcoef','invalid level value','msg'); error(' * ');
end
```

```
% Extract detail coefficients.
k = rmax   n;
first = sum(l(1:k-1))+1;
last = first+l(k)-1;
d = c(first:last);
```

71. function [a,d] = dwt(x,arg2,arg3)

函数 DWT 用于单水平一维小波变换。

DWT 执行单水平一维小波分解。其中字符串参数(arg2)可用于不同得小波函数('wname',参见 WFILTERS) 或者指定的小波滤波器 (Lo _ D and Hi _ D)。

【例子】

(1) [CA,CD] = DWT(X,'wname')

计算逼近系数矢量 CA 和细节系数矢量 CD;其中字符串参数 'wname' 用于表示小波函数名.

(2) [CA,CD] = DWT(X,Lo _ D,Hi _ D)

用给定的小波滤波器计算小波分解的逼近系数矢量 CA 和细节系数矢量 CD.

输入参数 Lo _ D 是分解的低通滤波器而 Hi _ D 是分解的高通滤波器,Lo _ D 和 Hi _ D 必须具有同样长度。

如果 lx 为 信号(X)的长度,lf 为 滤波器的长度,则有,

length(CA) = length(CD) = FLOOR((lx+lf-1)/2).

对于不同得信号扩展模式(signal extension modes),参见 DWTMODE.

参见 DWTMODE, DWTPER, IDWT, WAVEDEC, WAVEINFO.

【程序】

```
global DWT _ Ext _ Mode

% Check arguments.
if errargn('dwt',nargin,[2:3],nargout,[0:2]), error(' * '), end
if nargin - - 2
        [LoF _ D,HiF _ D] = wfilters(arg2,'d');
else
        LoF _ D = arg2; HiF _ D = arg3;
end

if      isempty(DWT _ Ext _ Mode) | DWT _ Ext _ Mode=='zpd' % zpd is the default
        % Decomposition.
        a = dyaddown(conv(x,LoF _ D));
        d = dyaddown(conv(x,HiF _ D));

elseif  DWT _ Ext _ Mode=='sym'

        % Symmetrization.
        lx = length(x);
        lf = length(LoF _ D);
        if size(x,1)==1
            x = [x(lf:-1:1) x x(lx:-1:lx-lf+1)];
        else
```

```
        x = [x(lf:-1:1); x ; x(lx:-1:lx-lf+1)];
    end

    % Decomposition.
    la = floor((lx+lf-1)/2);
    a = wkeep(dyaddown(conv(x,LoF_D)),la);
    d = wkeep(dyaddown(conv(x,HiF_D)),la);

elseif DWT_Ext_Mode=='spd'

    % Smooth padding.
    lx = length(x);
    lf = length(LoF_D);
    if size(x,1)==1
        x = [zeros(1,lf) x zeros(1,lf)];
    else
        x = [zeros(lf,1); x ; zeros(lf,1)];
    end
    for k=lf:-1:1
        x(k) = 2*x(k+1)-x(k+2);
    end
    for k=lf+lx+1:lx+2*lf
        x(k) = 2*x(k-1)-x(k-2);
    end

    % Decomposition.
    la = floor((lx+lf-1)/2);
    a = wkeep(dyaddown(conv(x,LoF_D)),la);
    d = wkeep(dyaddown(conv(x,HiF_D)),la);

else
    errargt('dwt','Invalid Extension Mode for DWT!','msg');
    error('*')

end
```

72. function [a,d] = dwtper(x,arg2,arg3)

函数 DWTPER 用于计算单水平离散一维小波变换(周期性的)(periodized).

【例子】

(1) [CA,CD] = DWTPER(X,'wname')

通过对信号矢量(X)周期性的小波分解(periodized wavelet decomposition),计算逼近系数矢量 CA 和细节系数矢量 CD;其中字符串参数 'wname' 用于表示小波函数名.

(2) [CA,CD] = DWTPER(X,Lo_D,Hi_D),

通过给定得滤波器亦可计算逼近系数矢量 CA 和细节系数矢量 CD。其中输入参数 Lo_D 是分解的低通

滤波器而 Hi _ D 是分解的高通滤波器,Lo _ D 和 Hi _ D 必须具有同样长度。如果 lx 为信号矢量(X) 的长度,则

length(CA) = length(CD) = CEIL(lx/2).

参见 DWT, IDWTPER.

【程序】

```
% Check arguments.
if errargn('dwtper',nargin,[2:3],nargout,[0:2]), error('*'), end
if nargin == 2
        [LoF _ D,HiF _ D,LoF _ R,HiF _ R] = wfilters(arg2);
else
        LoF _ D = arg2; HiF _ D = arg3;
end

% Periodization.
lx = length(x);
lf = length(LoF _ D);
if rem(lx,2) , x(lx+1) = x(lx); lx = lx+1; end
if size(x,1) ==1
        x = [x(lx-lf+1:lx) x x(1:lf)];
else
        x = [x(lx-lf+1:lx);x;x(1:lf)];
end

% Decomposition.
lp = ceil(lx/2);
a = wkeep(dyaddown(conv(x,LoF _ D)),lp);
d = wkeep(dyaddown(conv(x,HiF _ D)),lp);
```

73. function y = dyaddown(x,arg2,arg3)

函数 DYADDOWN 计算二分的下采样(Dyadic downsampling)

【例子】

(1) Y = DYADDOWN(X,EVENODD),

在此 X 为一矢量,返回的矢量 Y 是长度为矢量 X 二分下采样的结果矢量;整数参数 EVENODD 决定它是包含 X 矢量中的偶数下标元素(even- indexed samples)还是奇数下标元素(odd-indexed samples)。

如果 EVENODD 是偶数,则 Y(k) = X(2k).

如果 EVENODD 是奇数,则 Y(k) = X(2k-1).

(2) Y = DYADDOWN(X) 等于 Y = DYADDOWN(X,0)

(3) Y = DYADDOWN(X,EVENODD,'type') or

  Y = DYADDOWN(X,'type',EVENODD)

在此 X 是一个矩阵,返回值 Y 是 X 经压缩列(或行,或行和列),根据字符串'type' 是取 'c'(column,列)(或取 'r' (row, 行)或者 'm')而定。参数 EVENODD 的意义同前。

(4) Y = DYADDOWN(X) 就等于 Y = DYADDOWN(X,0,'c').

(5) Y = DYADDOWN(X,'type') 就等于 Y = DYADDOWN(X,0,'type').

(6) Y = DYADDOWN(X,EVENODD) 就等于 Y = DYADDOWN(X,EVENODD,'c').

说明实例： $X = \begin{vmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 9 & 0 \end{vmatrix}$，则 DYADDOWN(X,'c') = $\begin{vmatrix} 2 & 4 \\ 4 & 8 \\ 6 & 0 \end{vmatrix}$

DYADDOWN(X,'r',1) = $\begin{vmatrix} 1\ 2\ 3\ 4 \\ 3\ 6\ 9\ 0 \end{vmatrix}$，DYADDOWN(X,'m',1) = $\begin{vmatrix} 1\ 3 \\ 3\ 9 \end{vmatrix}$

参见 DYADUP.

【程序】

```
% Check arguments.
if errargn('dyaddown',nargin,[1:3],nargout,[0:1]), error('*'), end

[r,c] = size(x);
if min(r,c)<=1
        if     nargin==1, p = 0;
        elseif nargin==2, p = arg2;
        else errargt('dyaddown','too many arguments','msg');
error('*');
        end
        if rem(p,2), y = x(1:2:length(x)); else, y = x(2:2:length(x)); end
else
        if     nargin==1, o = 'c'; p = 0;
        elseif nargin==2
             if isstr(arg2), p = 0; o = lower(arg2(1));
             else, p = arg2; o = 'c';
             end
        else
             if isstr(arg2), p = arg3; o = lower(arg2(1));
             else, p = arg2; o = lower(arg3(1));
             end
        end
        if     o=='c'
             if rem(p,2), y = x(:,1:2:c);
             else, y = x(:,2:2:c);
             end
        elseif o=='r'
             if rem(p,2), y = x(1:2:r,:);
             else, y = x(2:2:r,:);
             end
        elseif o=='m'
             if rem(p,2), y = x(1:2:r,1:2:c);
             else, y = x(2:2:r,2:2:c);
             end
        else
```

```
            errargt('dyaddown','invalid argument value','msg');
error('*');
        end
end
```

74. function y = dyadup(x,arg2,arg3)

函数 DYADUP 计算二分的上采样(Dyadic upsampling),它通过简单填零而使该过程在小波重构算法中非常有用。

【例子】

(1) Y = DYADUP(X,EVENODD)

在此 X 为一矢量,返回的矢量 Y 是长度为矢量 X 两倍的上采样结果矢量;整数参数 EVENODD 决定它是在 X 矢量中的偶数下标元素(even-indexed samples)还是奇数下标元素(odd-indexed samples)位置中填零。

如果整数参数 EVENODD 是偶数,则 Y(2k−1)= X(k), Y(2k) = 0.

如果整数参数 EVENODD 是奇数,则 Y(2k−1)= 0 , Y(2k) = X(k).

(2) Y = DYADUP(X) 就等于 Y = DYADUP(X,1)

(3) Y = DYADUP(X,EVENODD,'type') 或

Y = DYADUP(X,'type',EVENODD)

在此 X 是一个矩阵,返回值 Y 是 X 经扩充列(或行,或行和列),根据字符串'type' 是取 'c'(column,列)(或取 'r' (row, 行)或者'm')而定。参数 EVENODD 的意义同前。

(4) Y = DYADUP(X) 就等于 Y = DYADUP(X,1,'c')

(5) Y = DYADUP(X,'type') 就等于 Y = DYADUP(X,1,'type')

(6) Y = DYADUP(X,EVENODD) 就等于 Y =DYADUP(X,EVENODD,'c')

说明实例 : $X = \begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix}$, 则 $DYADUP(X,'c') = \begin{vmatrix} 0 & 1 & 0 & 2 & 0 \\ 0 & 3 & 0 & 4 & 0 \end{vmatrix}$

$DYADUP(X,'r',0) = \begin{vmatrix} 1 & 2 \\ 0 & 0 \\ 3 & 4 \end{vmatrix}$, $DYADUP(X,'m',0) = \begin{vmatrix} 1 & 0 & 2 \\ 0 & 0 & 0 \\ 3 & 0 & 4 \end{vmatrix}$

参见 DYADDOWN.

【程序】

```
% Check arguments.
if errargn('dyadup',nargin,[1:3],nargout,[0:1]), error('*'), end

[r,c] = size(x);
mi = min(r,c);
dim = 1;
if mi>1
        dim = 2;
else
        if      nargin==2
                if isstr(arg2) , dim =2; end
        elseif nargin==3
                if isstr(arg2) | isstr(arg3) , dim = 2; end
        end
end
if dim==1
```

```
        if      nargin==1, p = 1;
        elseif nargin==2, p = arg2;
        else errargt('dyadup','too many arguments','msg'); error('*');
        end
        if rem(p,2)
            l = 2*length(x)+1; y = zeros(1,l); y(2:2:l) = x;
            if r>1, y = y'; end
        else
            l = 2*length(x)-1; y = zeros(1,l); y(1:2:l) = x;
            if r>1, y = y'; end
        end
else
        if      nargin==1, p = 1; o = 'c';
        elseif nargin==2
            if isstr(arg2), p = 1; o = lower(arg2(1));
            else p = arg2; o = 'c';
            end
        else
            if isstr(arg2), p = arg3; o = lower(arg2(1));
            else, p = arg2; o = lower(arg3(1));
            end
        end
        if      o=='c'
            if rem(p,2)
              nc = 2*c+1; y = zeros(r,nc); y(:,2:2:nc) = x;
            else
              nc = 2*c-1; y = zeros(r,nc); y(:,1:2:nc) = x;
            end
        elseif o=='r'
            if rem(p,2)
              nr = 2*r+1; y = zeros(nr,c); y(2:2:nr,:) = x;
            else
              nr = 2*r-1; y = zeros(nr,c); y(1:2:nr,:) = x;
            end
        elseif o=='m'
            if rem(p,2)
                    nr = 2*r+1; nc = 2*c+1;
                    y = zeros(nr,nc); y(2:2:nr,2:2:nc) = x;
            else
                    nr = 2*r-1; nc = 2*c-1;
                    y = zeros(nr,nc); y(1:2:nr,1:2:nc) = x;
            end
        else
            errargt('dyadup','invalid argument value','msg');
```

```
error('*');
        end
end
```

75. function x = idwt(a,d,arg3,arg4,arg5)

函数 IDWT 计算单水平逆离散一维小波变换。它根据不同小波基函数('wname'，参见 WFILTERS)，执行单水平逆离散一维小波重构(reconstruction)

【例子】

(1) X = IDWT(CA,CD,'wname')

根据逼近和细节系数(CA 和 CD)，返回单水平重构逼近系数矢量 X。字符串参数 'wname' 用来选择不同得小波基。

(2) X = IDWT(CA,CD,Lo _ R,Hi _ R)

用给定的重构低通滤波器(Lo _ R)和重构高通滤波器 (Hi _ R)及 逼近和细节系数(CA 和 CD)来计算单水平重构逼近系数矢量 X。

注意:Lo _ R 和 Hi _ R 必须是具有相同长度的矢量。

如果 la = (CA)的长度 = (CD) 的长度,lf 为 重构高通滤波器的长度,则(X)的长度为(2 * la－lf－2).

(3) X = IDWT(CA,CD,'wname',L)或 X =IDWT(CA,CD,Lo _ R,Hi _ R,L)

返回的是return the length－L central portion of the result

obtained using IDWT(CA,CD,'wname'). L 必须小于等于(2 * la－lf + 2)

参见 DWT，IDWTPER，UPWLEV.

【程序】

```
% Check arguments.
if errargn('idwt',nargin,[3:5],nargout,[0:1]), error('*'), end

if isstr(arg3)
        [LoF _ R,HiF _ R] = wfilters(arg3,'r');
        if nargin==4 , l = arg4; else , l = 2 * length(a)－length(LoF _ R)－2;
end
else
        LoF _ R = arg3; HiF _ R = arg4;
        if nargin==5 , l = arg5; else , l = 2 * length(a)－length(LoF _ R)+2;
end
end

% Reconstruction.
x = conv(dyadup(a),LoF _ R)+conv(dyadup(d),HiF _ R);
x = wkeep(x,l);
```

76. function x = idwtper(a,d,arg3,arg4,arg5)

用于计算单水平离散一维小波变换(周期性的)(periodized).

函数 IDWTPER 用于计算单水平逆离散一维小波变换(周期性的)(periodized).

【例子】

(1) X = IDWTPER(CA,CD,'wname')

通过对信号矢量(X)周期性的逆小波分解(periodized inverse wavelet transformation)，根据逼近系数矢量 CA 和细节系数矢量 CD,计算单水平重构逼近系数矢量 X。其中字符串参数 'wname' 用于表示小波函数名(参见 WFILTERS).

(2) X = IDWTPER(CA,CD,Lo _ R,Hi _ R)

不用小波基函数,也可用滤波器来计算单水平重构逼近系数矢量 X。其中 输入参数 Lo _ D 是重构的低通滤波器而 Hi _ D 是重构的高通滤波器,Lo _ D 和 Hi _ D 必须具有同样长度。如果

la = (CA)的长度 = length(CD)的长度,则(X)的长度 = 2 * la.

(3) X = IDWTPER(CA,CD,'wname',L) 或 X = IDWTPER(CA,CD,Lo _ R,Hi _ R,L)

在此,L 为结果矢量得长度,其余参数意义同前。

参见 DWTPER.

【程序】

```
% Check arguments.
if errargn('idwtper',nargin,[3:5],nargout,[0:1]), error(' * '), end

la = length(a);
if isstr(arg3)
        [LoF _R,HiF _R] = wfilters(arg3,'r');
        if nargin==4 , lx = arg4; else , lx = 2 * la; end
else
        LoF _R = arg3; HiF _R = arg4;
        if nargin==5 , lx = arg5; else , lx = 2 * la; end
end

% Reconstruction.
lf = length(LoF _ R);
lp = ceil(lx/2);
lm = (floor((lx+lf-1)/2) - lp)/2;
ar = ceil(lm);
al = floor(lm);
if size(a,1)==1
        a = [a(lp-al+1:lp) a a(1:ar)];
        d = [d(lp-al+1:lp) d d(1:ar)];
else
        a = [a(lp-al+1:lp);a;a(1:ar)];
        d = [d(lp-al+1:lp);d;d(1:ar)];
end
x = conv(dyadup(a),LoF _ R)+conv(dyadup(d),HiF _ R);
x = wkeep(x,lx);
```

77. function thr = thrwda1d(c,l,tptr,scal,n)

函数 THRWDA1D 用小波变换计算一维自动去噪的阈值。

【例子】

THR = THRWDA1D(C,L,TPTR,SCAL,N)

计算可用于小波自动去噪的阈值。

其中参数:

N 表示不同的小波分解水平;

[C,L]表示小波分解结构;

TPTR 则用于决定阈值选择规则;

SCAL 用于定义不同的去噪标度：

'one' 定义为无标度；

'sln' 定义为基于第一水平系数矢量噪声水平单个估计的标度；

'mln' 定义为再标度，可用于不同水平的噪声水平估计。

这样，所得结果为 THR(i)，即表示水平 i 的阈值。

参见 WDEN.

【程序】

```
% Check arguments.
if errargn('thrwda1d',nargin,[5],nargout,[0:1]), error('*'), end
if errargt('thrwda1d',tptr,'str'), error('*'), end
if errargt('thrwda1d',scal,'str'), error('*'), end
if errargt('thrwda1d',n,'int'), error('*'), end

% Threshold rescaling coefficients.
if strcmp(scal,'one'), s = ones(1,n);
elseif strcmp(scal,'sln'), s = ones(1,n) * wnoisest(c,l,1);
elseif strcmp(scal,'mln'), s = wnoisest(c,l,1:n);
else
errargt('thrwda1d','invalid argument value','msg'); error('*')
end

% Thresholds.
first = cumsum(l)+1; first = fliplr(first(1:length(first)-2));
ld = l(length(l)-1:-1:2); last = first+ld-1;
thr = zeros(1,n);
for k = 1:n
        flk = first(k):last(k);
        if tptr =='sqtwolog' | tptr=='minimaxi'
              t = thselect(c,tptr);
        else
              if s(k) < sqrt(eps) * max(c(flk))
                     t = 0;
              else
                     t = thselect(c(flk)/s(k),tptr);
              end
        end                    % threshold.
        thr(k) = t * s(k);  % rescaled threshold.
end
```

78. function thr = thselect(x,tptr)

函数 THSELECT 可用于选择去噪的阈值.

【例子】

THR = THSELECT(X,TPTR)

计算适合信号矢量 X 的阈值，其中字符串参数 TPTR 用来定义选择的规则。可用的规则为：

a) TPTR = 'rigrsure'，表示根据 Stein 的无偏风险估计(Stein's Unbiased Risk Estimate)原则，选择自适

应阈值。

b) TPTR = 'heursure'，表示根据首选自推理变化原则选择自适应阈值。(heuristic variant of the first option.)

c) TPTR = 'sqtwolog'，阈值为 sqrt(2 * log(length(X))).

d) TPTR = 'minimaxi'，极小极大阈值选择原则(minimax thresholding)。

注意：上述阈值选择规则阶皆为在分析信号中只含白噪声的模型之上，即，y = f(t) + e，其中 e 为白噪声（e～N(0,1)).对于含非白噪声的一维分析信号，当用重再表度(rescaling)的阈值参数 THR(参见 WDEN 函数中的 SCAL 参数)。

参见 WDEN.

【程序】

```
if errargn('thselect',nargin,[2],nargout,[0 1]), error('*'), end


n = length(x);
if strcmp(tptr,'rigrsure')
        sx2 = sort(abs(x)).^2;
        risks = (n-(2*(1:n))+(cumsum(sx2)+(n-1:-1:0).*sx2))/n;
        [risk,best] = min(risks);
        thr = sqrt(sx2(best));
elseif strcmp(tptr,'heursure')
        hthr = sqrt(2*log(n));
        eta = (norm(x).^2-n)/n;
        crit = (log(n)/log(2))^(1.5)/sqrt(n);
        if eta < crit
                thr = hthr;
        else
                thr = min(thselect(x,'rigrsure'),hthr);
        end
elseif strcmp(tptr,'sqtwolog')
        thr = sqrt(2*log(n));
elseif strcmp(tptr,'minimaxi')
        if n <= 32, thr = 0;
        else thr = 0.3936 + 0.1829*(log(n)/log(2)); end
else
        errargt('thselect','invalid argument value','msg');
        error('*')
end
```

79. function y = upcoef(o,x,arg3,arg4,arg5,arg6)

函数 UPCOEF 可用于用一维小波系数来直接重构分析信号。

【例子】

(1) Y = UPCOEF(O,X,'wname',N)

计算矢量 X 的 N 步重构系数。'wname' 为字符串变量参数，用于选择小波基函数。N 必须是一个正整数。如果 O = 'a'，表示重构逼近系数，如果 O = 'd'，表示重构细节系数(detail coefficients)。

(2) Y = UPCOEF(O,X,'wname',N,L)

计算矢量 X 的 N 步重构系数，并得到结果矢量中距中心长度为 L 的部分的矢量。

(3) Y = UPCOEF(O,X,Lo _ R,Hi _ R,N) or

Y = UPCOEF(O,X,Lo _ R,Hi _ R,N,L)

不用小波基函数,也可用滤波器来计算重构系数矢量 X。其中 输入参数 Lo _ D 是重构低通滤波器而 Hi _ D是重构高通滤波器。

(4) Y = UPCOEF(O,X,'wname') 等于

Y = UPCOEF(O,X,'wname',1).

(5) Y = UPCOEF(O,X,Lo _ R,Hi _ R) 等于

Y = UPCOEF(O,X,Lo _ R,Hi _ R,1).

在(4)和(5)的情况中,Y 是没有被截断的.

参见 IDWT.

【程序】

```
% Check arguments.
if errargn('upcoef',nargin,[3:6],nargout,[0:1]), error('*'), end

o = lower(o(1));
if isstr(arg3)
        [LoF _ R,HiF _ R] = wfilters(arg3,'r');
        if nargin < 4, n = 1; l = 0;
        else
                n = arg4;
                if nargin < 5, l = 0; else l = arg5; end
        end
else
        LoF _ R = arg3; HiF _ R = arg4;
        if nargin < 5, n = 1;
        else
                n = arg5;
                if nargin < 6, l = 0; else l = arg6; end
        end
end
if (n<1) | (n~=fix(n))
        errargt('upcoef','invalid argument value','msg'); error('*');
end
if (l<0) | (l~=fix(l))
        errargt('upcoef','invalid argument value','msg'); error('*');
end

if  o=='a'
        % Approximation reconstruction.
        F1 = LoF _ R; F2 = F1;
elseif o=='d'
        % Detail reconstruction.
        F1 = HiF _ R; F2 = LoF _ R;
else
```

```
        errargt('upcoef','invalid argument value','msg'); error('*');
end
y = conv(dyadup(x),F1);
for  k=2:n
        y = conv(dyadup(y),F2);
end
if l, y = wkeep(y,l); end
```

80. function [c,l,a] = upwlev(c,l,arg3,arg4)

函数 UPWLEV 用于一维信号小波分解的单水平重构。

【例子】

(1) [NC,NL,CA] = UPWLEV(C,L,'wname')

执行一维信号小波分解的单水平重构，从已得的 [C,L] 计算出新的水平的 [NC,NL]，同时还得到最近的逼近系数矢量 CA。

如果[C,L] 是水平 n(= length(L)−2)，则 [NC,NL] 是同样的分解矢量，但其水平为 n−1，而 CA 则是水平 n 的逼近系数矢量。

字符串参数'wname' 表示所选小波基函数名。

C 为原始小波分解矢量，而 L 为相应得簿记矢量(bookkeeping vector)，其详细信息参见函数 WAVEDEC。

(2) [NC,NL,CA] = UPWLEV(C,L,Lo_R,Hi_R)，不用小波基函数，也可用滤波器来计算上述重构矢量。其中输入参数 Lo_D 是重构低通滤波器而 Hi_D 是重构高通滤波器。

参见 IDWT，UPCOEF，WAVEDEC.

【程序】

```
% Check arguments.
if errargn('upwlev',nargin,[3:4],nargout,[0:3]), error('*'), end
if nargin == 3
        [LoF_R,HiF_R] = wfilters(arg3,'r');
else
        LoF_R = arg3; HiF_R = arg4;
end

% Extract last approximation.
ll = length(l);
l1 = l(1);
a = c(1:l1);

% One step reconstruction of the wavelet decomposition structure.
if ll > 2
        l2 = l(2);
        l3 = l(3);
        l = l(2:ll);          l(1) = l3;

        d = c(l1+1:l1+l2);
        lc = length(c);
        ra = idwt(a,d,LoF_R,HiF_R,l3);
```

```
        c = c(l1+l2+1-l3:lc);
        c(1:l3) = ra;
else
        l = [];
        c = [];
end
```

81. function [c,l] = wavedec(x,n,arg3,arg4)

函数 WAVEDEC 可用于计算多水平一维小波分解。它通过或是小波基函数或是小波分解滤波器来计算多水平一维小波分解

【例子】

(1) [C,L] = WAVEDEC(X,N,'wname')

返回分析信号水平 N 的关于指定小波函数的小波分解。其中 N 必须是正整数,所得结果包括小波分解矢量 C 和簿记矢量 L.

(2) [C,L] = WAVEDEC(X,N,Lo _ D,Hi _ D),小波分解也可用分解低通滤波器 Lo _ D 和分解高通滤波器 Hi _ D 来求得。

小波分解的结构可表达为:

C = [app. coef. (N)|det. coef. (N)|… |det. coef. (1)]

L(1) = length of app. coef. (N)

L(i) = length of det. coef. (N-i+2) for i = 2,…,N+1

L(N+2) = length(X).

参见 DWT, WAVEINFO, WFILTERS, WMAXLEV.

【程序】

```
% Check arguments.
if errargn('wavedec',nargin,[3:4],nargout,[0:2]), error(' * '), end
if errargt('wavedec',n,'int'), error(' * '), end
if nargin==3
        [LoF _ D,HiF _ D] = wfilters(arg3,'d');
else
        LoF _ D = arg3; HiF _ D = arg4;
end

% Initialization.
s = size(x); x = x(:)';                 % row vector
c = []; l = [length(x)];

for k = 1:n
        [x,d] = dwt(x,LoF _ D,HiF _ D); % decomposition
        c = [d c];                      % store detail
        l = [length(d) l];              % store length
end

% Last approximation.
c = [x c];
```

```
l = [length(x) 1];

if s(1)>1, c = c'; l = l'; end
```

82. function thr = wdcbm(c,l,alpha,m)

函数 WDCBM 用 Birge-Massart 策略对一维分析信号去噪和压缩

【例子】

(1) THR = WDCBM(C,L,ALPHA,M)

用 Birge-Massart 策略的小波系数选择规则得到不同水平的阈值以用于去噪和压缩。其中[C,L] 是分析信号的小波分解结构。

字符串参数 ALPHA 必须是大于1的实数。一般说来，当 ALPHA = 1.5时表示压缩，而当 ALPHA = 3时，表示去噪。参数 M 是一个正整数，当 M=L(1)/2时，最粗糙的逼近系数矢量的长度大于2。

(2) WDCBM(C,L,ALPHA) 就等于 WDCBM(C,L,ALPHA,L(1)/2)。

参见 WDEN，WDENCMP，WPDENCMP.

【程序】

```
% Check arguments.
if errargn('wdcbm',nargin,[2:4],nargout,[0 1]), error('*'), end
if errargt('wdcbm',alpha-1,'rep'), error('*'), end
if nargin==4
        if errargt('wdcbm',m,'int'), error('*'), end
else
        m = l(1)/2;             % the number of detail coefs at level j0-1.
end
j0 = length(l)-2;               % low frequency cutoff.

% Wavelet coefficients selection.
for j=1:j0
        % number of coefs to be kept.
        n = m/(j0+1-j)^alpha;
        n = min(round(n),l(j0-j+2));
        % threshold.
        if n == l(j0-j+2)
                thr(j) = 0;
        else
                d = detcoef(c,l,j);
                d = sort(abs(d));
                thr(j) = d(length(d)-n);
        end
end
```

83. function [xd,cxd,lxd] = wden(in1,in2,in3,in4,in5,in6,in7)

函数 WDEN 用小波来自动对一维分析信号进行去噪。

【例子】

(1) [XD,CXD,LXD] = WDEN(X,TPTR,SORH,SCAL,N,'wname')

通过对小波系数矢量取阈值而得到去噪了的分析信号矢量；附加的输出矢量[CXD,LXD]则为去噪了的分析信号矢量的小波分解结构。字符串参数 TPTR 则用于决定阈值选择规则：

TPTR 可用的规则为：

a) TPTR = 'rigrsure'，表示根据 Stein 的无偏风险估计(Stein's Unbiased Risk Estimate)原则，选择自适应阈值。

b) TPTR = 'heursure'，表示根据首选自推理变化原则选择自适应阈值。heuristic variant of the first option.

c) TPTR = 'sqtwolog'，阈值为 sqrt(2 * log(length(X)))。

d) TPTR = 'minimaxi'，极小极大阈值选择原则(minimax thresholding)。

(参见 THSELECT 函数，可得更详细信息)。

字符串参数 SORH ('s' or 'h') 用以决定是用软或硬的去噪模式(参见 WTHRESH 函数，可得更详细信息)。

字符串参数 SCAL 用于定义不同的去噪标度：

'one' 定义为无标度；

'sln' 定义为基于第一水平系数矢量噪声水平单个估计的标度；

'mln' 定义为再标度，可用于不同水平的噪声水平估计。

参数 N 表示不同的小波分解水平；

字符串参数'wname'则用于定义不同正交小波基函数。

(2) [XD,CXD,LXD] = WDEN(C,L,TPTR,SORH,SCAL,N,'wname')

直接用小波分解结构[C,L]，而不用分析信号本身 X，也可得到同前的结果。

参见 THSELECT, WAVEDEC, WDENCMP, WFILTERS, WTHRESH.

【程序】

```
% Check arguments.
if nargin==7
        if errargn('wden',nargin,[7],nargout,[0:3])
                error('*')
        end
        c=in1; l=in2; tptr=in3; sorh=in4; scal=in5; n=in6; w=in7;
else
        if errargn('wden',nargin,[6],nargout,[0:3]), error('*'), end
        x=in1; tptr=in2; sorh=in3; scal=in4; n=in5; w=in6;
end
if errargt('wden',tptr,'str'), error('*'), end
if errargt('wden',sorh,'str'), error('*'), end
if errargt('wden',scal,'str'), error('*'), end
if errargt('wden',n,'int'), error('*'), end
if errargt('wden',w,'str'), error('*'), end

if nargin==6
        % Wavelet decomposition of x.
        [c,l] = wavedec(x,n,w);
end

% Threshold rescaling coefficients.
if strcmp(scal,'one'), s = ones(1,n);
elseif strcmp(scal,'sln'), s = ones(1,n) * wnoisest(c,l,1);
```

```
elseif strcmp(scal,'mln'), s = wnoisest(c,l,1:n);
else
errargt('wden','invalid argument value','msg'); error('*')
end

% Wavelet coefficients thresholding.
first = cumsum(l)+1;
first = fliplr(first(1:length(first)-2));
ld = l(length(l)-1:-1:2);
last = first+ld-1;
cxd = c;
lxd = l;
for k = 1:n
        flk = first(k):last(k);
        if tptr=='sqtwolog' | tptr=='minimaxi'
                thr = thselect(c,tptr);
        else
                if s(k) < sqrt(eps) * max(c(flk))
                        thr = 0;
        else
                        thr = thselect(c(flk)/s(k),tptr);
                end
        end                                     % threshold.
        thr = thr * s(k);                       % rescaled threshold.
        cxd(flk) = wthresh(c(flk),sorh,thr);    % thresholding or shrinking.
end

% Wavelet reconstruction of xd.
xd = waverec(cxd,lxd,w);
```

84. function [xc,cxc,lxc,perf0,perfl2] = …
wdencmp(o,in2,in3,in4,in5,in6,in7,in8)

函数 WDENCMP 可用于用小波进行去噪和压缩。它可对一分析信号或图像来进行去噪和压缩。

【例子】

(1) [XC,CXC,LXC,PERF0,PERFL2] = WDENCMP('gbl',X,'wname',N,THR,SORH,KEEPAPP)

该函数通过对小波系数矢量取全局正阈值 THR，而得到去噪了的分析信号矢量或矩阵 XC 及附加的输出矢量[CXD,LXD]，它们为去噪了的分析信号矢量或矩阵的小波分解结构，字符串参数 TPTR 则用于决定阈值选择规则；另两个输出参数 PERFL2和 PERF0分别表示2范数回收(recovery)和压缩率，其中：

PERFL2 = 100 * (vector-norm of CXC/vector-norm of C) ^ 2

式中的 [C,L] 表示分析信号的小波分解结构。

输入参数：

X 为原始数据；

'wname' 为所选小波函数名；

N 为分解水平；

THR 为全局正阈值；

字符串参数 SORH 为除噪模式，可能取值为：'s' or 'h'；

如果参数 KEEPAPP = 1，则表示逼近系数不被取阈值，其余则 可能。

(2) WDENCMP('gbl',C,L,W,N,THR,SORH,KEEPAPP)

与上例效果同，唯一不同之点在于本调用没有用原始数据 X，而是用它的小波分解结构 [C,L]。

(3) 对一维信号而言，如采用'lvd' 选择而不是'gbl'时，如下面两例：

WDENCMP('lvd',X, 'wname',N,THR,SORH) or

WDENCMP('lvd',C,L, 'wname',N,THR,SORH)

也可得前述两例的结果，此时输入参数 THR 必须为一矢量，即包含了所有水平所需阈值，此外，还保持了逼近功能。

(4) 对二维信号而言，如采用'lvd' 选择而不是'gbl'时，如下面两例，：

WDENCMP('lvd',X, 'wname',N,THR,SORH) or

WDENCMP('lvd',C,L, 'wname',N,THR,SORH)

可得前述例子的相同结果，但此时输入参数 THR 必须为一矩阵(3×N)，即包含了三维方向的所有水平所需阈值。

参见 DDENCMP, WAVEDEC, WAVEDEC2, WDEN, WPDENCMP, WTHRESH.

【程序】

```
% Check arguments and set problem dimension.
dim = 1;            % initialize dimension to 1D.
if o == 'gbl'
        if errargn('wdencmp',nargin,[7:8],nargout,[0:1 3:5]), error(' * '),
end
        if nargin==7, x=in2; w=in3; n=in4; thr=in5; sorh=in6; keepapp=in7;
                if min(size(x))~=1, dim = 2; end
        else, c=in2; l=in3; w=in4; n=in5; thr=in6; sorh=in7;
keepapp=in8;
                if min(size(l))~=1, dim = 2; end
        end
elseif o == 'lvd'
        if errargn('wdencmp',nargin,[6:7],nargout,[0:1 3:5]), error(' * '),
end
        if nargin==6, x=in2; w=in3; n=in4; thr=in5; sorh=in6;
                if min(size(x))~=1, dim = 2; end
        else, c=in2; l=in3; w=in4; n=in5; thr=in6; sorh=in7;
                if min(size(l))~=1, dim = 2; end
        end
else, errargt('wdencmp','invalid argument value','msg'); error(' * ')
end
if errargt('wdencmp',w,'str'), error(' * '), end
if errargt('wdencmp',n,'int'), error(' * '), end
if errargt('wdencmp',thr,'re0'), error(' * '), end
if errargt('wdencmp',sorh,'str'), error(' * '), end

% Wavelet decomposition of x (if not given).
if (o=='gbl' & nargin==7) | (o=='lvd' & nargin==6)
```

```
        if dim == 1,              [c,l] = wavedec(x,n,w);
        else,                     [c,l] = wavedec2(x,n,w);
        end
end

% Wavelet coefficients thresholding.
if o=='gbl'
        if keepapp
            % keep approximation.
            cxc = c;
            if dim == 1, inddet = l(1)+1:length(c);
            else, inddet = prod(l(1,:))+1:length(c); end
            % threshold detail coefficients.
            cxc(inddet) = wthresh(c(inddet),sorh,thr);
        else    % threshold all coefficients.
                cxc = wthresh(c,sorh,thr);
        end
else
        if dim == 1, cxc = wthcoef('t',c,l,1:n,thr,sorh);
        else
                cxc = wthcoef2('h',c,l,1:n,thr(1,:),sorh);
                cxc = wthcoef2('d',cxc,l,1:n,thr(2,:),sorh);
                cxc = wthcoef2('v',cxc,l,1:n,thr(3,:),sorh);
        end
end
lxc = l;

% Wavelet reconstruction of xd.
if dim == 1,          xc = waverec(cxc,lxc,w);
else                  xc = waverec2(cxc,lxc,w);
end

if nargout<4 , return; end

% Compute compression score.
perf0 = 100 * (length(find(cxc==0))/length(cxc));
if nargout<5 , return; end

% Compute L^2 recovery score.
perfl2 = 100 * ((cxc * cxc')/(c * c'));
```

85. function [out1,out2,out3,out4,out5,out6,out7,out8] = wfilters(wname,o)

函数 WFILTERS 计算小波滤波器(Wavelet filters).

【例子】

(1) [LO_D,HI_D,LO_R,HI_R] = WFILTERS('wname')

计算不同正交或双正交的小波函数的4个滤波器，

此4个滤波器为：

LO _ D，分解低通滤波器；

HI _ D，分解高通滤波器；

LO _ R，重构低通滤波器；

HI _ R，重构高通滤波器。

可用的小波函数为：

```
%      Daubechies ：'db1' or 'haar'，'db2'，… ，'db50'
%        Coiflets      ：'coif1'，…，      'coif5'
%        Symlets       ：'sym2' ，…'sym8'
%        Biorthogonal：'bior1.1'，'bior1.3' ，'bior1.5'
%                      'bior2.2'，'bior2.4' ，'bior2.6'，'bior2.8'
%                      'bior3.1'，'bior3.3' ，'bior3.5'，'bior3.7'
%                      'bior3.9'，'bior4.4'，'bior5.5'，'bior6.8'.
```

(2) [F1,F2] = WFILTERS('wname','type')

可用于计算两个滤波器：

如果 'type' = 'd' (分解滤波器)，则得到 LO _ D and HI _ D

如果 'type' = 'r' (重构滤波器)，则得到 LO _ R and HI _ R

如果 'type' = 'l' (低通滤波器)，则得到 LO _ D and LO _ R

如果 'type' = 'h' (高通滤波器)，则得到 HI _ D and HI _ R。

参见 BIORFILT，ORTHFILT，WAVEINFO.

【程序】

```
% Check arguments.
if errargn('wfilters',nargin,[1 2],nargout,[0 1 2 4 8]), error(' * '); end
if errargt('wfilters',wname,'str') , error(' * '), end

wname          = deblankl(wname);
[wtype,fname] = wavemngr('info',wname);
mat _ f         = findstr('.mat',fname);
if mat _ f
  err = 0;
  eval(['load ' fname],'err = 1;');
  if err
      msg = ['invalid wavelet file ; ' fname];
      errargt('wfilters',msg,'msg');
      error(' * ');
      return;
  end
end

if     wtype==1                 % orth. wavelet
      if ~isempty(mat _ f)
                F = eval(wname);
      else
```

```
                F = feval(fname,wname);
        end
        [LoF_D,HiF_D,LoF_R,HiF_R] = orthfilt(F);

elseif wtype == 2                    % biorth. wavelet
        if isempty(mat_f)
                [Rf,Df] = feval(fname,wname);
        else
                if exist('Rf')~=1 | exist('Df')~=1
                        msg = ['invalid biorthogonal wavelet file : ' fname];
                        errargt('wfilters',msg,'msg');
                        error('*');
                end
        end
        [LoF_D,HiF_D1,LoF_R1,HiF_R,LoF_D2,HiF_D,LoF_R,HiF_R2] = ...
                                biorfilt(Df,Rf,1);
        out5 = LoF_D2;          out6 = HiF_D1;
        out7 = LoF_R1;          out8 = HiF_R2;
end

if nargin==1
        out1 = LoF_D; out2 = HiF_D; out3 = LoF_R; out4 = HiF_R;
else
        o = lower(o(1));
        if      o=='d' , out1 = LoF_D; out2 = HiF_D;
        elseif  o=='r' , out1 = LoF_R; out2 = HiF_R;
        elseif  o=='l' , out1 = LoF_D; out2 = LoF_R;
        elseif  o=='h' , out1 = HiF_D; out2 = HiF_R;
        else
                errargt('wfilters','invalid argument value','msg');     error('*');
        end
end
```

86. function y = wkeep(x,arg2,arg3)

函数 WKEEP 可用于保存一个矢量或一个矩阵的一部分。

【例子】

(1) 对于矢量分析信号 X：Y = WKEEP (X，L，O)

将得到矢量 Y，其中 L 为其长度。字符串参数 O 可取 'c' 'l' 和 'r' 分别表示 Y 是取自矢量 X 的中部、左边部分和右边部分。

(2) Y = WKEEP(X,L) 就等于 Y = WKEEP(X,L,'c')。

(3) 对于矩阵信号：Y = WKEEP(X,S)

将得到新矩阵 Y，表示取自矩阵 X 的中部，其中参数 S 表示 Y 的大小。

【程序】

```
% Check arguments.
if errargn('wkeep',nargin,[2:3],nargout,[0:1]), error('*'), end
```

```
[r,c] = size(x);
y = x;
if min(r,c)<=1
        lx = length(x); l = arg2;
        if (l>=0) & (l<lx) & (l == fix(l))
                if nargin==2, o = 'c'; else , o = lower(arg3(1)); end
                if o=='c'
                        d = (lx-l)/2; y = y(1+floor(d):lx-ceil(d));
                elseif o=='l', y = y(1:l);
                elseif o=='r' , y = y(lx-l+1:lx);
                else
                        errargt('wkeep','invalid argument','msg');error('*');
                end
        end
else
        sx = size(x); s = arg2(:)';
        K = find(s>sx);
        s(K) =sx(K);
        m = find((s < 0) | (s ~= fix(s)));
        if isempty(m)
                ss = (sx-s)/2;
                first = 1+floor(ss);
                last = sx-ceil(ss);
                y = y(first(1):last(1),first(2):last(2));
        end
end
```

87. function [x,xn] = wnoise(num,n,snrat,init)

函数 WNOISE 可用于产生一个含噪声的小波去噪的检测矢量。

【例子】

(1) X = WNOISE(NUM,N)

将得到一个检测矢量。参数 NUM 表示不同的常用于小波分析中的知名函数：

NUM = 1　　Blocks 函数

NUM = 2　　Bumps 函数

NUM = 3　　Heavy sine 函数

NUM = 4　　Doppler 函数

NUM = 5　　Quadchirp 函数

NUM = 6　　Mishmash 函数

该检测矢量 X 的长度为 2^N。

(2) [X,XN] = WNOISE(NUM,N,SNRAT)

将得到一个检测矢量 X，其标准偏差为：std(X) = SNRAT，另一个矢量 XN 为一个在矢量 X 上附加了一个均值为零、方差为1的高斯白噪声，即 N(0,1)，这样，矢量 XN 的信噪比(signal-to-noise ratio)就是 SNRAT.

(3) [X,XN] = WNOISE(NUM,N,SNRAT,INIT)

将得到前述两个检测矢量 X 和 XN，新增参数 INIT 表示产生随机矢量 XN 的种子(seed).

参见 WDEN.

【程序】

```
if nargin==2
if errargn('wnoise',nargin,[2],nargout,[0 1]), error('*'), end
else
if errargn('wnoise',nargin,[3:4],nargout,[0:2]), error('*'), end
end

t=[0.1  0.13  0.15  0.23  0.25  0.40  0.44  0.65  0.76  0.78  0.81];
h=[4  -5  3  -4  5  -4.2  2.1  4.3  -3.1  5.1  -4.2];
if num==1          % blocks.
        tt=linspace(0,1,2^ n); x=zeros(1,length(tt));
        for j=1:11
                x = x + ( h(j) * (1+sign(tt-t(j)))/2 );
        end
elseif num==2      % bumps.
        h=abs(h);
        w=0.01*[0.5 0.5 0.6 1 1 3 1 1 0.5 0.8 0.5];
        tt=linspace(0,1,2^ n); x=zeros(1,length(tt));
        for j= 1:11
                x = x + ( h(j) ./ (1+ ((tt-t(j))/w(j)).^ 4));
        end
elseif num==3      % heavy sine.
        x=linspace(0,1,2^ n);
        x = 4*sin(4*pi*x) - sign(x-0.3) - sign(0.72-x);
elseif num==4      % doppler.
        x=linspace(0,1,2^ n);
        epsil = 0.05;
        x = sqrt(x.*(1-x)).* sin( 2*pi*(1+epsil)./(x+epsil) );
elseif num==5      % quadchirp.
        t=linspace(0,1,2^ n);
        x = sin( (pi/3) *t .* (2^ n * t.^ 2) );
elseif num==6      % mishmash.
        t=linspace(0,1,2^ n);
        x = sin( (pi/3) *t .* (2^ n * t.^ 2) );
        x = x + sin( pi* (2^ n* 0.6902) * t );
        x = x + sin( pi* t .* (2^ n* 0.125* t) );
else
        errargt('wnoise','invalid argument value','msg');
        error('*')
end

if nargin>=3
```

```
        x = (x * (snrat/std(x)));
        if nargin==4, randn('seed',init); end
        xn = x + randn(1,2^n);
end
```

88. function stdc = wnoisest(c,l,s)

函数 WNOISEST 可用于估计一维小波系数矢量的噪声。

【例子】

STDC = WNOISEST(C,L,S)

计算细节系数矢量的标准偏差，

[C,L] 为小波分解结构(参见 WAVEDEC 函数).

所用估计子(estimator)是最大绝对偏差(Maximum Absolute Deviation/0.6745),适合于一维信号零均高斯白噪声(参见 THSELECT 函数).

参见 THSELECT, WAVEDEC, WDEN.

【程序】

```
% Check arguments.
if errargn('wnoisest',nargin,[3],nargout,[0:1]), error('*'), end

rmax = length(l);          nmax = rmax-2;
first = cumsum(l)+1;
first = fliplr(first(1:length(first)-2));
ld = l(rmax-1:-1:2);
last = first+ld-1;

for k = 1:length(s)
        flk = first(s(k)):last(s(k));
        stdc(k) = median(abs(c(flk)))/0.6745;
end
```

89. function x = wrcoef(o,c,l,arg4,arg5,arg6)

函数 WRCOEF 用于从一维小波分解结构(C 和 L),或特定小波基函数('wname'),或特定重构滤波器(Lo_R and Hi_R)进行重构系数矢量计算。

【例子】

(1) X = WRCOEF('type',C,L,'wname',N)

表示基于小波分解结构[C,L](参见 WAVEDEC),及相应的小波基函数'wname',计算重构系数矢量 X。其中参数 N 表示水平;字符串参数'type' = 'a'表示逼近系数;而'type' = 'd'表示细节系数矢量。

注意:当'type' = 'a'时, N 可以为0, 对于其他别的情况,N 一定得是正整数,而且 N <= length(L)-2.

(2) X = WRCOEF('type',C,L,Lo_R,Hi_R,N)

表示用重构高通滤波器(Hi_R)和重构低通滤波器(Lo_R)同样可计算重构系数矢量 X。其余参数同前。

(3) X = WRCOEF('type',C,L,'wname') 及 X = WRCOEF('type',C,L,Lo_R,Hi_R)

表示所得的重构系数矢量 X,具有最高水平数 N =(L)的长度-2。

参见 APPCOEF, DETCOEF, WAVEDEC.

【程序】

```
% Check arguments.
if errargn('wrcoef',nargin,[4:6],nargout,[0:1]), error('*'), end
o = lower(o(1));
```

```
rmax = length(l);          nmax = rmax-2;
if o=='a', nmin = 0; else , nmin = 1; end
if isstr(arg4)
        [LoF_R,HiF_R] = wfilters(arg4,'r');
        if nargin == 5, n = arg5; else, n = nmax; end
else
        LoF_R = arg4; HiF_R = arg5;
        if nargin == 6, n = arg6; else, n = nmax; end
end
if (n<nmin) | (n>nmax) | (n~=fix(n))
        errargt('wrcoef','invalid level value','msg'); error('*');
end

imin = rmax-n;
if o=='a'
      % Extract approximation.
      x = appcoef(c,l,LoF_R,HiF_R,n);

        % Approximation reconstruction.
        for k=1:n
        s = l(imin+k);
                x = wkeep(conv(dyadup(x),LoF_R),s);
        end

elseif o=='d'
     % Extract detail coefficients.
     x = detcoef(c,l,n);

     % Detail reconstruction.
   s = l(imin+1);
     x = wkeep(conv(dyadup(x),HiF_R),s);

        for k=2:n
        s = l(imin+k);
                x = wkeep(conv(dyadup(x),LoF_R),s);
        end

else
        errargt('wrcoef','invalid argument value','msg'); error('*');
end
```

90. function m = wrmcoef(o,c,l,arg4,arg5,arg6)

函数 WRMCOEF 用于从一维小波分解结构得到重构系数矢量矩阵。

【例子】

(1) M = WRMCOEF(O,C,L,W,N)

表示基于小波分解结构[C,L]（参见 WAVEDEC），及相应的小波基函数'w'，计算重构系数矢量矩阵 M。其中参数 N 表示水平；字符串参数'type' = 'a'表示逼近系数；而'type' = 'd'表示细节系数矢量。

注意：

1）当'type' = 'a'时，N 可以为0，对于其他别的情况，N 一定得是正整数，而且 N <= length(L)－2.

2）所得矩阵 M 为重构系数矢量矩阵，它是按行来进行排列的。

(2) M = WRMCOEF(O,C,L,Lo,Hi,N)

表示用重构高通滤波器(Hi)和重构低通滤波器(Lo)同样可计算重构系数矢量矩阵 M。其余参数同前。

(3) M = WRMCOEF (O，C，L，W) 或

M = WRMCOEF (O，C，L，Lo，Hi)

表示重构系数矢量矩阵 M 含有所有可能水平的重构系数矢量。

参见 APPCOEF，DETCOEF，WRCOEF，WAVEDEC.

【程序】

```
% Check arguments.
if errargn ('wrmcoef', nargin, [4: 6], nargout, [0: 1]), error ('*'), end
o = lower (o (1));
rmax = length (l);            nmax = rmax-2;
if o=='a', nmin = 0; else , nmin = 1; end
if isstr (arg4)
        [LoF_R, HiF_R] = wfilters (arg4,'r');
        if nargin == 5, n = arg5; else, n = [nmin: nmax]; end
else
        LoF_R = arg4; HiF_R = arg5;
        if nargin == 6, n = arg6; else, n = [nmin: nmax]; end
end
if find ( (n<nmin) | (n>nmax) | (n~=fix (n)))
        errargt ('wrmcoef','invalid level (s) value (s)','msg'); error ('*');
end

% Initialization
if size (l, 1) >1 , c = c'; l = l'; end
m = zeros (length (n), l (rmax));

if o=='a'
        for p = nmax: -1: 0
                [c, l, a] = upwlev (c, l, LoF_R, HiF_R);
                j = find (p==n);
                if j
                        % Approximation reconstruction.
                imin = length (l) -p;
                        for k=1: p
                s = l (imin+k);
                                a = wkeep (conv (dyadup (a), LoF_R), s);
                        end
                        a = a (ones (size (j)),:);
```

```
                m (j,:) = a;
            end
        end

elseif o=='d'
        for p = 1: nmax
            j = find (p==n);
            if j
                % Extract detail coefficients.
                d = detcoef (c, l, p);

                %Detail reconstruction.
              imin = rmax-p;
              s = l (imin+1);
                d = wkeep (conv (dyadup (d), HiF_R), s);
                for k=2: p
                  s = l (imin+k);
                      d = wkeep (conv (dyadup (d), LoF_R), s);
                end
                d = d (ones (size (j)),:);
                m (j,:) = d;
            end
        end

else
        errargt ('wrmcoef','invalid argument value','msg'); error ('*');
end
```

91. function y = wthresh(x,sorh,t)

函数 WTHRESH 执行软(soft)或硬(hard)取阈值.

Y = WTHRESH(X,SORH,T) 表示对分析信号矢量 X 或分析信号矩阵 X 进行以参数 T 为阈值的软取阈值(如果 SORH = 's')或硬取阈值 (如果 SORH = 'h') 运算。

【例子】

(1) Y = WTHRESH(X,'s',T)

所得结果为 Y = SIGN(X).(|X|-T)+，执行软取阈值运算。

(2) Y = WTHRESH(X,'h',T)

所得结果为 Y = X.1_(|X|>T),)+，执行硬取阈值运算。

参见 WDEN，WDENCMP，WPDENCMP.

【程序】

```
% Check arguments.
if errargn('wthresh',nargin,[3],nargout,[0:1]), error('*'), end

if sorh=='s'
        tmp = (abs(x)-t); tmp = (tmp+abs(tmp))/2;
        y = sign(x).*tmp;
```

```
elseif sorh = = 'h'
        y = x. * (abs(x)>t);
else
        errargt('wthresh','invalid argument value','msg'), error(' * ')
end
```

92. function [a1, a2, a3] = simuff (p, w1, b1, f1, w2, b2, f2, w3, b3, f3)

函数 SIMUFF 用于模拟前传网络。SIMUFF 函数最高可模拟三层网络。

【例子】

(1) SIMUFF(P,W1,B1,'F1',...,Wn,Bn,'Fn')

其中参数：

P 为包含输入列矢量的矩阵；

Wi 为包含第 i 层的权重矩阵；

Bi 为包含第 i 层的偏置列矢量；

字符串参数 Fi 为第 i 层的转换函数；

该函数返回第 n 层的输出。

(2) [A1,A2,...] = SIMUFF(P,W1,B1,'F1',...,Wn,Bn,'Fn')

输出参数 Ai 为第 i 层的输出。

(3) 使用示例：[w1,b1,w2,b2] = initff([0 10; -5 5],3,'tansig',2,'purelin');

p = [2; -3];

a = simuff(p,w1,b1,'tansig',w2,b2,'purelin')

参见 INITFF，TRAINBPX，TRAINLM.

【程序】

```
if all([4 7 10] ~= nargin),error('Wrong number of input arguments'),end

if nargin == 4
  a1 = feval(f1,w1 * p,b1);

elseif nargin == 7
  a1 = feval(f1,w1 * p,b1);
  a2 = feval(f2,w2 * a1,b2);
  if nargout <= 1, a1 = a2; end

elseif nargin == 10
  a1 = feval(f1,w1 * p,b1);
  a2 = feval(f2,w2 * a1,b2);
  a3 = feval(f3,w3 * a2,b3);
  if nargout <= 1, a1 = a3; end
end
```

93. function [w1,b1,w2,b2,w3,b3] = initff(p,s1,f1,s2,f2,s3,f3)

函数 INITFF Inititialize 对三层及以下的前传网络进行初始化。

【例子】

(1) [W1,B1,…] = INITFF(P,S1,'F1',…,Sn,'Fn')

输入参数：

P 为包含输入参数矢量的 R×2阶矩阵，matrix of input vectors. Si 为第 i 层的大小；

字符串参数 Fi 为第 i 层的转换函数。

输出参数：

Wi 为第 i 层的权重矩阵；

Bi 为第 i 层的偏置列矢量。

(2) [W1,B1,…] = INITFF(P,S1,'F1',…,T,'Fn')

其中：T 为包含目标矢量的 SnxQ 阶矩阵。将同样返回权重矩阵和偏置矢量。

注意：P 矩阵的每一行将必须包含第 i 层的输入的最小和最大值。

(3) 使用实例：[W1,b1,W2,b2] = initff([0 10; - 5 5],5,'tansig',3,'purelin');

p = [4; - 1];

a = simuff(p,W1,b1,'tansig',W2,b2,'purelin')

参见 SIMUFF，TRAINBPX，TRAINLM

【程序】

```
if all([3 5 7] ~= nargin),error('Wrong number of input arguments'),end

[R,Q] = size(p);
if (Q < 2), error('P must contain at least two elements in each row.'),end

if nargin == 3
  [S,Q] = size(s1);
if max(S,Q) > 1, s1 = S; end
  [w1,b1] = feval(feval(f1,'init'),s1,p);

elseif nargin == 5
  [w1,b1] = feval(feval(f1,'init'),s1,p);
  x = ones(s1,1) * feval(f1,'output');
  [S,Q] = size(s2);
  if max(S,Q) > 1, s2 = S; end
  [w2,b2] = feval(feval(f2,'init'),s2,x);

elseif nargin == 7
  [w1,b1] = feval(feval(f1,'init'),s1,p);
  x = ones(s1,1) * feval(f1,'output');
  [w2,b2] = feval(feval(f2,'init'),s2,x);
  x = ones(s2,1) * feval(f2,'output');
  [S,Q] = size(s3);
  if max(S,Q) > 1, s3 = S; end
  [w3,b3] = feval(feval(f3,'init'),s3,x);
end
```

94. function [a,b,c,d,e,f,g,h] = trainbp(i,j,k,l,m,n,o,p,q,r,s,t)

函数 TRAINBP 用误差反传来训练前传网络，TRAINBP 函数可调用权重和偏置来训练三层前传网络。

【例子】

[W1,B1,W2,B2,…,TE,TR] = TRAINBP(W1,B1,F1,W2,B2,F2,…,P,T,TP)

输入参数：

Wi 为第 i 层的(SixR)阶的权重矩阵；

Bi 为第 i 层的(S1x1)阶的偏置列矢量；

字符串参数 Fi 为第 i 层的转换函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(SxQ)阶的目标矩阵；

TP 为训练参数（可选择）。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25.

TP(2)为训练的最大数目，预置值为100.

TP(3)为平方误差和目标，预置值为 0.02.

TP(4)为学习率（Learning rate），预置值为0.01.

没给训练参数的都将采用预置值。

参见 INITFF，SIMFF，TRAINBPX，TRAINLM.

【程序】

```
if all ( [5 6 8 9 11 12] ~= nargin), error ('Wrong number of input arguments'), end

if nargin == 5
   [a, b, c, d] = tbp1 (i, j, k, l, m);
elseif nargin == 6
   [a, b, c, d] = tbp1 (i, j, k, l, m, n);
elseif nargin == 8
   [a, b, c, d, e, f] = tbp2 (i, j, k, l, m, n, o, p);
elseif nargin == 9
   [a, b, c, d, e, f] = tbp2 (i, j, k, l, m, n, o, p, q);
elseif nargin == 11
   [a, b, c, d, e, f, g, h] = tbp3 (i, j, k, l, m, n, o, p, q, r, s);
elseif nargin == 12
   [a, b, c, d, e, f, g, h] = tbp3 (i, j, k, l, m, n, o, p, q, r, s, t);
end
```

95. function [a,b,c,d,e,f,g,h] = trainbpx(i,j,k,l,m,n,o,p,q,r,s,t)

函数 TRAINBPX 采用快速误差反传来训练前传网络，可以训练一层、二层、最高可以训练到三层网络。

【例子】

[W1,B1,W2,B2,…,TE,TR] = TRAINBPX(W1,B1,F1,W2,B2,F2,…,P,T,TP)

输入参数：

Wi 为第 i 层的权重矩阵；

Bi 为第 i 层的偏置列矢量；

字符串参数 Fi 为第 i 层的转换函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(SxQ)阶的目标矩阵；

TP 为训练参数（可选择）。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为100；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为学习率(Learning rate)，预置值为0.01；

TP(5)为学习率增量，预置值为 1.05；

TP(6)为学习率减量，预置值为 0.7；

TP(7)为惯量常数，预置值为 0.9；

TP(8)为最大误差率(Maximum error ratio)，预置值为1.04。

没给训练参数的都将采用预置值。

参见 NNTRAIN，BACKPROP，INITFF，SIMFF，TRAINBP，TRAINLM.

【程序】

```
if all([5 6 8 9 11 12] ~= nargin),error('Wrong number of input arguments'),end

if nargin == 5
  [a,b,c,d] = tbpx1(i,j,k,l,m);
elseif nargin == 6
  [a,b,c,d] = tbpx1(i,j,k,l,m,n);
elseif nargin == 8
  [a,b,c,d,e,f] = tbpx2(i,j,k,l,m,n,o,p);
elseif nargin == 9
  [a,b,c,d,e,f] = tbpx2(i,j,k,l,m,n,o,p,q);
elseif nargin == 11
  [a,b,c,d,e,f,g,h] = tbpx3(i,j,k,l,m,n,o,p,q,r,s);
elseif nargin == 12
  [a,b,c,d,e,f,g,h] = tbpx3(i,j,k,l,m,n,o,p,q,r,s,t);
end
```

96. function [a,b,c,d,e,f,g,h] = trainbpm(i,j,k,l,m,n,o,p,q,r,s,t)

函数 TRAINBPM 采用误差反传加惯量常数来训练前传网络，可以训练一层、二层、最高可以训练到三层网络。

【例子】

[W1,B1,W2,B2,…,TE,TR] = TRAINBPM(W1,B1,F1,W2,B2,F2,…,P,T,TP)

输入参数：

Wi 为第 i 层的权重矩阵；

Bi 为第 i 层的偏置列矢量；

字符串参数 Fi 为第 i 层的转换函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(SNxQ)阶的目标矩阵；

TP 为训练参数（可选择)。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为学习率(Learning rate)，预置值为0.01；

TP(5)为学习率增量，预置值为 1.05；

TP(6)为学习率减量，预置值为 0.7；

TP(7)为惯量常数，预置值为 0.9；

TP(8)为最大误差率(Maximum error ratio)，预置值为1.04。

没给训练参数的都将采用预置值，

参见 NNTRAIN，BACKPROP，INITFF，SIMFF，TRAINBPX，TRAINLM.

【程序】

```
if all([5 6 8 9 11 12] ~= nargin),error('Wrong number of input arguments'),end

if nargin == 5
  [a,b,c,d] = tbpx1(i,j,k,l,m,[NaN NaN NaN NaN 1 1 NaN NaN]);
  d = d(1,:);

elseif nargin == 6
  n = [n NaN+zeros(1,8) 1];
  len = length(n);
  n = n([1 2 3 4 len len 5 6]);
  [a,b,c,d] = tbpx1(i,j,k,l,m,n);
  d = d(1,:);

elseif nargin == 8
  [a,b,c,d,e,f] = tbpx2(i,j,k,l,m,n,o,p,...
    [NaN NaN NaN NaN 1 1 NaN NaN]);
  f = f(1,:);

elseif nargin == 9
  q = [q NaN+zeros(1,8) 1];
  len = length(q);
  q = q([1 2 3 4 len len 5 6]);
  [a,b,c,d,e,f] = tbpx2(i,j,k,l,m,n,o,p,q);
  f = f(1,:);

elseif nargin == 11
  [a,b,c,d,e,f,g,h] = tbpx3(i,j,k,l,m,n,o,p,q,r,s,...
```

```
    [NaN NaN NaN NaN 1 1 NaN NaN]);
  h = h(1,:);

elseif nargin == 12
  t = [t NaN+zeros(1,8) 1];
  len = length(t);
  t = t([1 2 3 4 len len 5 6]);
  [a,b,c,d,e,f,g,h] = tbpx3(i,j,k,l,m,n,o,p,q,r,s,t);
  h = h(1,:);
end
```

97. function [a,b,c,d,e,f,g,h] = trainlm(i,j,k,l,m,n,o,p,q,r,s,t)

函数 TRAINLM 采用 Levenberg-Marquardt 方法来训练前传网络，可以训练一层、二层、最高可以训练到三层网络。

【例子】

[W1,B1,W2,B2,...,TE,TR] = TRAINLM(W1,B1,F1,W2,B2,F2,…,P,T,TP)

输入参数：

Wi 为第 i 层的权重矩阵；

Bi 为第 i 层的偏置列矢量；

字符串参数 Fi 为第 i 层的转换函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(SxQ)阶的目标矩阵；

TP 为训练参数（可选择)。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为最小梯度，预置值为0.0001；

TP(5)为 MU 的初始值，预置值为0.001；

TP(6)为 MU 增量的乘数，预置值为 10；

TP(7)为 MU 减量的乘数，预置值为 0.1；

TP(8)为 MU 的最大值，预置值为1^{10}。

没给训练参数的都将采用预置值。

参见 NNTRAIN，BACKPROP，INITFF，SIMFF，TRAINBP，TRAINBPX.

【程序】

```
if all([5 6 8 9 11 12] ~= nargin),error('Wrong number of input arguments'),end

if nargin == 5
  [a,b,c,d] = tlm1(i,j,k,l,m);
elseif nargin == 6
```

```
  [a,b,c,d] = tlm1(i,j,k,l,m,n);
elseif nargin == 8
  [a,b,c,d,e,f] = tlm2(i,j,k,l,m,n,o,p);
elseif nargin == 9
  [a,b,c,d,e,f] = tlm2(i,j,k,l,m,n,o,p,q);
elseif nargin == 11
  [a,b,c,d,e,f,g,h] = tlm3(i,j,k,l,m,n,o,p,q,r,s);
elseif nargin == 12
  [a,b,c,d,e,f,g,h] = tlm3(i,j,k,l,m,n,o,p,q,r,s,t);
end
```

98. function n = purelin(n,b)

函数 PURELIN 为线性转换或称活性函数

【例子】

(1) PURELIN (N)

输入参数：N 为包含网络输入矢量的 (SxQ) 阶矩阵。

应用实例：n = −10：0.1：10；

a = purelin (n)；

plot (n，a)

(2) PURELIN (Z，B) 用于批处理情况。

输入参数：

Z 为网络加权了的输入矢量的 (SxQ) 阶矩阵；

B 为偏置矢量的 (Sx1) 列矢量。

返回值为网络矩阵中在−1至1之间的数。

(3) PURELIN('delta') 返回 delta 函数的名称。

PURELIN('init') 返回初始函数的名称。

PURELIN('name') 返回该转换或活性函数的全名。

PURELIN('output') 返回该函数值的输出范围。

参见 NNTRANS，SATLIN，SATLINS.

【程序】

```
if nargin < 1, error('Not enough arguments.'); end

if isstr(n)
  if strcmp(lower(n),'delta')
    n = 'deltalin';
  elseif strcmp(lower(n),'init')
    n = 'rands';
  elseif strcmp(lower(n),'name')
    n = 'Linear';
  elseif strcmp(lower(n),'output')
    n = [-1e100 1e100];
  else
    error('Unrecognized property.')
  end
else
```

```
  if nargin==2
    [nr,nc] = size(n);
    n = n + b * ones(1,nc);
  end
end
```

99. function a = tansig(n,b)

函数 TANSIG 为双曲线正弦 S 形转换或称活性函数

【例子】

(1) TANSIG(N)

输入参数:N 为包含网络输入矢量的(SxQ)阶矩阵。该函数的返回值为矩阵 N 的在－1至1之间的数。

应用实例: n = －10:0.1:10;

a = tansig(n);

plot(n,a)

(2) TANSIG(Z,B) 用于批处理情况。

输入参数:

Z 为网络加权了的输入矢量的(SxQ)阶矩阵;

B 为偏置矢量的(Sx1)列矢量。

返回值为网络矩阵中在－1至1之间的数。

(3) TANSIG('delta') 返回 delta 函数的名称。

TANSIG('init') 返回初始函数的名称。

TANSIG('name') 返回该转换或活性函数的全名。

TANSIG('output') 返回该函数值的输出范围。

参见 NNTRANS, BACKPROP, NWTAN, LOGSIG.

【程序】

```
if nargin < 1, error('Not enough arguments.'); end

if isstr(n)
  if strcmp(lower(n),'delta')
    a = 'deltatan';
  elseif strcmp(lower(n),'init')
    a = 'nwtan';
  elseif strcmp(lower(n),'name')
    a = 'Hyperbolic Tangent Sigmoid';
  elseif strcmp(lower(n),'output')
    a = [-1 1];
  else
    error('Unrecognized property.')
  end
else
  if nargin==2
    [nr,nc] = size(n);
    n = n + b * ones(1,nc);
  end
  a = 2./(1 + exp(-2*n)) - 1;
```

```
  i = find( ~finite(a));
  a(i) = sign(n(i));
end
```

100. function a = logsig(n,b)

函数 LOGSIG 为对数 S 形(sigmoid)转换函数。

【例子】

(1) LOGSIG(N)

输入参数:N 为包含网络输入矢量的(SxQ)阶矩阵。该函数的返回值为矩阵 N 的在0至1之间的数。

应用实例: n = −10:0.1:10;

a = logsig(n);

plot(n,a)

(2) LOGSIG(Z,B) 用于批处理情况。

输入参数:

Z 为网络加权了的输入矢量的(SxQ)阶矩阵;

B 为偏置矢量的(Sx1)列矢量。

返回值为网络矩阵中在0至1之间的数。

(3) LOGSIG('delta') 返回 delta 函数的名称。

LOGSIG('init') 返回初始函数的名称。

LOGSIG('name') 返回该转换或活性函数的全名。

LOGSIG('output') 返回该函数值的输出范围。

参见 NNTRANS, BACKPROP, NWTAN, LOGSIG.

【程序】

```
if nargin < 1, error('Note enough arguments.'); end

if isstr(n)
  if strcmp(lower(n),'   delta   ')
    a = 'deltalog';
  elseif strcmp(lower(n),'   init   ')
    a = 'nwlog';
  elseif strcmp(lower(n),'name   ')
    a = 'Log Sigmoid';
  elseif strcmp(lower(n),'output   ')
    a = [0 1];
  else
    error('Unrecognized property.')
  end
else
  if nargin==2
    [nr,nc] = size(n);
    n = n + b * ones(1,nc);
  end
  a = 1 ./ (1+exp(-n));
  i = find(~finite(a));
  a(i) = sign(n(i)) * 0.5 + 0.5;
```

```
end
```

101. function d = deltalin(a,d,w)

函数 DELTALIN PURELIN 神经元的 Delta 函数。

【例子】

(1) DELATLIN(A)

A 为包含输出矢量的(S1xQ)阶矩阵。

该函数返回包含输出矢量通过对 PURELIN 转换函数的(S1xQ)阶导数矩阵。

(2) DELTALIN(A,E)

E 为(S1xQ)阶误差矩阵。

该函数返回一个(S1xQ)阶的输出层的误差导数矩阵。

(3) DELTALIN(A,D,W)

输入参数:

D 为(S2xQ)阶含下一层 delta 矢量的矩阵。

W 为(S2xS1)阶的层间的权重矩阵。

该函数返回一个(S1xQ)阶的隐蔽层的误差导数矩阵。

参见 NNTRANS, BACKPROP, PURELIN.

【程序】

```
if nargin == 1
   d = ones(size(a));
elseif nargin == 2
  d = d;
else
  d = w' * d;
end
```

102. function d = deltatan(a,d,w)

函数 DELTATAN TANSIG 神经元的 Delta 函数.

【例子】

(1) DELTATAN(A)

A 为包含输出矢量的(S1xQ)阶矩阵。

该函数返回包含输出矢量的导数矩阵。

(2) DELTATAN(A,E)

输入参数:

A 为包含输出矢量的(S1xQ)阶矩阵;

E 为(S1xQ)阶误差矩阵。

该函数返回一个(SxQ)阶的输出层的误差导数矩阵。

(3) DELTATAN(A,D,W)

输入参数:

D 为(S2xQ)阶含下一层 delta 矢量的矩阵;

W 为(S2xS1)阶的层间的权重矩阵。

该函数返回一个(S1xQ)阶的隐蔽层的误差导数矩阵。

参见 NNTRANS, BACKPROP, TANSIG, DELTALOG, DELTALIN.

【程序】

```
if nargin < 1,error('Not enough input arguments'),end
```

```
if nargin == 1
  d = 1-(a. * a);
elseif nargin == 2
  d = (1-(a. * a)). * d;
else
  d = (1-(a. * a)). * (w' * d);
end
```

103. function d = deltalog(a,d,w)

函数 DELTALOG LOGSIG 神经元的 Delta 函数。

【例子】

(1) DELTALOG(A)

A 为包含输出矢量的(S1xQ)阶矩阵。

该函数返回包含输出矢量的导数矩阵。

(2) DELTALOG(A,E)

输入参数：

A 为包含输出矢量的(S1xQ)阶矩阵；

E 为(S1xQ)阶误差矩阵。

该函数返回一个(S1xQ)阶的输出层的误差导数矩阵。

(3) DELTATAN(A,D,W)

输入参数：

D 为(S2xQ)阶含下一层 delta 矢量的矩阵；

W 为(S2xS1)阶的层间的权重矩阵。

该函数返回一个(S1xQ)阶的隐蔽层的误差导数矩阵。

参见 NNTRANS, BACKPROP, LOGSIG, DELTALIN, DELTATAN

【程序】

```
if nargin < 1,error('Not enough input arguments'),end

if nargin == 1
  d = a. * (1-a);
elseif nargin == 2
  d = a. * (1-a). * d;
else
  d = a. * (1-a). * (w' * d);
end
```

104. function [dw,db] = learnbp(p,d,lr)

函数 LEARNBP 为误差反传学习规则函数。

【例子】

[dW,dB] = LEARNBP(P,D,LR)

输入参数：

P 为(RxQ)阶输入矢量矩阵。

E 为(SxQ)阶误差矢量矩阵。

LR 为学习率（预置值为 1).

输出参数：

dW 为权重增量矩阵.

dB 为偏置增量矩阵.

参见 NNLEARN, BACKPROP, SIMFF, INITFF, TRAINBP.

【程序】

```
if nargin < 2, error('Not enough arguments.'); end
if nargin == 3, d = lr * d; end

dw = d * p';
if nargout == 2
  [R,Q] = size(p);
  db = d * ones(Q,1);
end
```

105. function [dw,db] = learnbm(p,d,lr,mc,dw,db)

函数 LEARNBPM 为含有惯量因子的误差反传学习规则函数。

【例子】

[dW,dB] = LEARNBM(P,D,LR,MC,dW,dB)

输入参数:

P 为(RxQ) 阶输入矢量矩阵;

D 为(SxQ) 阶误差矢量矩阵;

lr 为学习率;

mc 为惯量因子常数;

dW 为(SxR)阶权重增量矩阵;

dB 为(Sx1)阶偏置增量矢量。

输出参数:

dW 为新权重增量矩阵;

dB 为新偏置增量矢量.

参见 NNLEARN, BACKPROP, SIMFF, INITFF, TRAINBPX.

【程序】

```
if nargin < 5,error('Not enough input arguments'),end

x = (1-mc) * lr * d;
dw = mc * dw + x * p';
if nargout == 2
  [R,Q] = size(p);
  db = mc * db + x * ones(Q,1);
end
```

106. function j = learnlm (p, d)

函数 LEARNLM 为 Levenberg-Marquardt 学习规则函数。

【例子】

LEARNLM(P,D)

P 为(RxQ) 阶输入矢量矩阵;

D 为(SxQ) 阶误差矢量矩阵。

返回:

偏 Jacobian 矩阵。

参见 NNLEARN, BACKPROP, INITFF, SIMFF, TRAINLM.

【程序】

```
if nargin < 2, error('Wrong number of arguments.'),end

[R,Q]=size(p);
[S,Q]=size(d);
j = nncpy(d',R) .* nncpyi(p',S);
```

107. function m = errsurf(p,t,wv,bv,f)

函数 ERRSURF 为误差表面函数.

【例子】

(1) ERRSURF(P,T,WV,BV,F)

P 为(1xQ)阶输入矢量矩阵;

T 为(1xQ)阶目标矢量矩阵;

WV 为权重的行矢量数值;

BV 为偏置的行矢量数值。

字符串参数:F 为转换或活性函数。

该函数返回在 WV 和 BV 域中的误差值。

(2) 应用实例:

```
p = [-6.0 -6.1 -4.1 -4.0 +4.0 +4.1 +6.0 +6.1];
t = [+0.0 +0.0 +.97 +.99 +.01 +.03 -1.0 +1.0];
wv = -1:.1:1; bv = -2.5:.25:2.5;
ES = errsurf(p,t,wv,bv,'logsig');
plotes(wv,bv,ES,[60 30])
```

参见 LINLAYER, BACKPROP, PLOTES, PLOTEP.

【程序】

```
if nargin < 5,error('Not enough input arguments.');end

m = zeros(length(bv),length(wv));
for Y=1:length(bv);
  m(Y,:) = nnsumc((t' * ones(1,length(wv))-feval(f,p' * wv +bv(Y))).^2);
end
```

108. function plotes(wv,bv,es,v)

函数 PLOTES 为误差表面作图函数。

【例子】

(1) PLOTES(WV,BV,ES,V)

WV 为权重的行矢量数值;

BV 为偏量的行矢量数值;

ES 为 WV 和 BV 域中的误差值,由 ERRSURF 函数算出;

V 为视点. 预量值为 [-37.5, 30].

(2) 应用实例:

```
p = [3 2];
t = [0.4 0.8];
wv = -4:0.4:4; bv = wv;
ES = errsurf(p,t,wv,bv,'logsig');
plotes(wv,bv,ES,[60 30])
```

参见 LINNET, BACKPROP, PLOTEP, ERRSURF.

【程序】

```
if nargin < 3, error('Not enough input arguments'),end

maxe = max(max(es));
mine = min(min(es));
drop = maxe--mine;
surfpos = mine - drop;
contpos = mine - drop * 0.95;

newplot;

% LEFT 3-D PLOT
% ==============

subplot(1,2,1);
[px,py] = gradient(es,wv,bv);
scolor = sqrt(px.^2+py.^2);

% SURFACE
sh = surf(wv,bv,es,scolor);
hold on
sh = surf(wv,bv,zeros(length(wv),length(bv))+surfpos,scolor);
set(sh,'edgecolor',[0.5 0.5 0.5])

% ERROR GOAL
if 0
minw = min(wv);
maxw = max(wv);
minb = min(bv);
maxb = max(bv);
z1 = plot3([minw maxw maxw minw minw],...
      [minb minb maxb maxb minb],[0 0 0 0 0],'w');
z2 = plot3([minw minw],[minb minb],[0 es(1,1)],'w');
z3 = plot3([maxw maxw],[minb minb],[0 es(1,length(wv))],'w');
z4 = plot3([maxw maxw],[maxb maxb],[0 es(length(bv),length(wv))],'w');
z5 = plot3([minw minw],[maxb maxb],[0 es(length(bv),1)],'w');
set([z1 z2 z3 z4 z5],'color',[1 1 0])
end

% TITLES
xlabel('Weight W');
ylabel('Bias B');
zlabel('Sum Squared Error')
title('Error Surface')
```

```
% WEIGHT & BIAS
set(gca,'xlim',[min(wv),max(wv)])
set(gca,'ylim',[min(bv),max(bv)])
zlim = get(gca,'zlim');

% VIEW
if nargin == 4, view(v), end
set(gca,'zlim',[surfpos maxe]);

% RIGHT 2-D PLOT
% ===============

subplot(1,2,2);

% SURFACE
sh = surf(wv,bv,es * 0,scolor);
hold on
set(sh,'edgecolor',[0.5 0.5 0.5])

% CONTOUR
[cc,ch] = contour(wv,bv,es,12);
hold off
set(ch,'edgecolor',[1 1 1])

% TITLES
xlabel('Weight W');
ylabel('Bias B');
title('Error Contour')

% VIEW
view([0 90])
set(gca,'xlim',[min(wv) max(wv)])
set(gca,'ylim',[min(bv) max(bv)])

% COLOR
colormap(cool);
```

109. function hh = plotep(w,b,e,h)

函数 PLOTEP 为权重和偏置在误差表面位置作图函数。

【例子】

(1) H = PLOTEP(W,B,E)

输入参数：

W 为当前权重值；

B 为当前偏置值；

E 为当前误差。

输出参数:H 为下一次调用函数所需信息。

(2) PLOTEP(W,B,E,H) 用于继续调用

输入参数:H 为前一次调用函数 PLOTEP 所得信息。

输出参数:H 为下一次调用函数所需信息。

参见 LINNET, BACKPROP, ERRSURF, PLOTES

【程序】

```
if nargin < 3, error('Not enough input arguments'),end

% GET LAST POSITION
% ==================

if nargin == 4
  w2 = h(1);
  b2 = h(2);
  delete(h(3:length(h)));
end

% MOVE MARKERS
% ==============
hold on
subplot(1,2,1);
zlim = get(gca,'zlim');
up = -zlim(1)*0.05;
plot3(w,b,zlim(1)+up,'.w','erasemode','none')
h2 = plot3([w w],[b b],[e zlim(1)]+up,'.w','markersize',20);

subplot(1,2,2);
h1 = plot(w,b,'.w','markersize',20);

% CONNECT NEW POSITION
% =====================

if nargin == 4
  subplot(1,2,1)
  plot3([w2 w],[b2 b],[0 0]+zlim(1)+up,'b','linewidth',2)

  subplot(1,2,2)
  plot([w w2],[b b2],'b','linewidth',2);
end

hold off
drawnow
```

```
if nargout == 1
  hh = [w; o; h1; h2];
end
```

110. function [w1,b1,i,tr] = tbp1(w1,b1,f1,p,t,tp,wv,bv,es,v)

函数 TBP1 用于训练一层前传误差反传网络。

【例子】

[W,B,TE,TR] = TBP1(W,B,F,P,T,TP)

输入参数：

W 为(SxR)阶权重矩阵；

B 为(Sx1)阶的偏置列矢量；

字符串参数 F 为 转换(或活性)函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(SxQ)阶的目标矩阵。

TP 为训练参数（可选择）.

输出参数：

W 为新权重；

B 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目,预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标,预置值为 0.02；

TP(4)为学习率(Learning rate)，预置值为0.01。

没给训练参数的都将采用预置值。

参见 NNTRAIN，BACKPROP，INITFF，SIMFF，TRAINBP，TRAINLM.

【程序】

```
if nargin < 5,error('Not enough arguments.'),end

% TRAINING PARAMETERS
if nargin == 5, tp = []; end
tp = nndef(tp,[25 1000 0.02 0.01]);
df = tp(1);
me = tp(2);
eg = tp(3);
lr = tp(4);
df1 = feval(f1,'delta');

% PRESENTATION PHASE
a1 = feval(f1,w1 * p,b1);
e = t-a1;
SSE = sumsqr(e);

% TRAINING RECORD
```

```
tr = zeros(1,me+1);
tr(1) = SSE;

% PLOTTING FLAG
[r,q] = size(p);
[s,q] = size(t);
plottype = (nargin >= 8) + (max(r,s) == 1);

% PLOTTING
newplot;
message = sprintf('TRAINBP: %%g/%g epochs, SSE = %%g.\n',me);
fprintf(message,0,SSE)
if plottype == 1
  h = plotfa(p,t,p,a1);
elseif plottype == 2
  if nargin <= 8, es = errsurf(p,t,wv,bv,f1); end
  if nargin <= 9, v = [-37.5 30]; end
  plotes(wv,bv,es,v);
  h = plotep(w1,b1,SSE);
else
  h = ploterr(tr(1),eg);
end

% TRAINING
for i=1:me

  % CHECK PHASE
  if SSE < eg, i = i - 1; break, end

  % BACKPROPAGATION PHASE
  d1 = feval(df1,a1,e);

  % LEARNING PHASE
  [dw1,db1] = learnbp(p,d1,lr);
  w1 = w1 + dw1; b1 = b1 + db1;

  % PRESENTATION PHASE
  a1 = feval(f1,w1 * p,b1);
  e = t-a1;
  SSE = sumsqr(e);

  % TRAINING RECORD
  tr(i+1) = SSE;
```

```
  % PLOTTING
  if rem(i,df) == 0
    fprintf(message,i,SSE)
    if plottype == 1
      delete(h);
      h = plot(p,a1);
    elseif plottype == 2
      h = plotep(w1,b1,SSE,h);
    else
      h = ploterr(tr(1:(i+1)),eg,h);
    end
  end
end

% TRAINING RECORD
tr = tr(1:(i+1));

% PLOTTING
if rem(i,df) ~= 0
  fprintf(message,i,SSE)
  if plottype == 1
    delete(h);
    plot(p,a1);
  elseif plottype == 2
    plotep(w1,b1,SSE,h);
  else
    ploterr(tr,eg,h);
  end
end

% WARNINGS
if SSE > eg
  disp('          ')
  disp('TRAINBP: Network error did not reach the error goal.')
  disp(' Further training may be necessary, or try different')
  disp(' initial weights and biases.')
  disp('          ')
end
```

111. function [w1,b1,w2,b2,i,tr]=tbp2(w1,b1,f1,w2,b2,f2,p,t,tp)

函数 TBP2 用于训练二层前传误差反传网络。

【例子】

[W1, B1, W2, B2, TE, TR] = TBP2 (W1, B1, F1, W2, B2, F2, P, T, TP)

输入参数：

Wi 为第 i 层的权重矩阵；

Bi 为第 i 层的的偏置列矢量；

字符串参数 Fi 为第 i 层的转换函数；

P 为包含输入矢量的（RxQ）阶的输入矩阵；

T 为包含目标矢量的（S2xQ）阶的目标矩阵；

TP 为训练参数（可选择）。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为学习率(Learning rate)，预置值为0.01。

没给训练参数的都将采用预置值。

【程序】

```
if nargin == 8, error('Not enough arguments.'), end

% TRAINING PARAMETERS
if nargin == 7, tp = []; end
tp = nndef(tp, [25 1000 0.02 0.01]);
df = tp (1);
me = tp (2);
eg = tp (3);
lr = tp (4);
df1 = feval(f1,'delta');
df2 = feval(f2,'delta');

% PRESENTATION PHASE
a1 = feval(f1, w1 * p, b1);
a2 = feval(f2, w2 * a1, b2);
e = t-a2;
SSE = sumsqr(e);

% PLOTTING FLAG
[r, q] = size(p);
[s2, q] = size(t);
plottype = max(r, s2) == 1;

% TRAINING RECORD
tr = zeros(1, me);
tr(1) = SSE;
```

```
% PLOTTING
newplot;
message = sprintf('TRAINBP: %%%g/%%g epochs, SSE = %%%g.\n', me);
fprintf(message, 0, SSE)
if plottype
  h = plotfa(p, t, p, a2);
else
  h = ploterr(tr (1), eg);
end

for i=1: me

  % CHECK PHASE
  if SSE < eg, i=i-1; break, end

  % BACKPROPAGATION PHASE
  d2 = feval(df2, a2, e);
  d1 = feval(df1, a1, d2, w2);

  % LEARNING PHASE
   [dw1, db1] = learnbp(p, d1, lr);
   [dw2, db2] = learnbp(a1, d2, lr);
  w1 = w1 - dw1; b1 = b1 + db1;
  w2 = w2 - dw2; b2 = b2 + db2;

  % PRESENTATION PHASE
  a1 = feval(f1, w1 * p, b1);
  a2 = feval(f2, w2 * a1, b2);
  e = t-a2;
  SSE = sumsqr (e);

  % TRAINING RECORD
  tr(i+1) = SSE;

  % PLOTTING
  if rem(i, df) == 0
  fprintf(message, i, SSE)
    if plottype
      delete(h);
      h = plot(p, a2);
      drawnow;
    else
      h = ploterr(tr (1: (i+1)), eg, h);
    end
```

```
  end
end

% TRAINING RECORD
tr = tr(1: (i+1));

% PLOTTING
if rem(i, df) ~= 0
  fprintf(message, i, SSE)
  if plottype
    delete(h);
    plot(p, a2);
    drawnow.
  else
    ploterr(tr, eg, h);
  end
end

% WARNINGS
if SSE > eg
  disp('     ')
  disp('TRAINBP: Network error did not reach the error goal.')
  disp('  Further training may be necessary, or try different')
  disp('  initial weights and biases and/or more hidden neurons.')
  disp('     ')
end
```

112. function [w1,b1,w2,b2,w3,b3,i,tr] = tbp3(w1,b1,f1,w2,b2,f2,w3,b3,f3,p,t,tp)

函数 TBP3 用于训练3层前传误差反传网络

【例子】

[W1,B1,W2,B2,W3,B3,TE,TR] = TBP3(W1,B1,F1,W2,B2,F2,W3,B3,F3,P,T,TP)

输入参数：

Wi 为第 i 层的(SixR)阶的权重矩阵；

Bi 为第 i 层的(Six1)阶的偏置列矢量；

字符串参数 Fi 为第 i 层的转换函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(S2xQ)阶的目标矩阵；

TP 为训练参数（可选择）。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为学习率(Learning rate)，预置值为0.01。

没给训练参数的都将采用预置值。

【程序】

```
if nargin < 11,error('Not enough arguments.'),end
% TRAINING PARAMETERS
if nargin == 11, tp = []; end
tp = nndef(tp,[25 1000 0.02 0.01]);
df = tp(1);
me = tp(2);
eg = tp(3);
lr = tp(4);
df1 = feval(f1,'delta');
df2 = feval(f2,'delta');
df3 = feval(f3,'delta');

% PRESENTATION PHASE
a1 = feval(f1,w1 * p,b1);
a2 = feval(f2,w2 * a1,b2);
a3 = feval(f3,w3 * a2,b3);
e = t-a3;
SSE = sumsqr(e);
tr = zeros(1,me);
tr(1) = SSE;

% PLOTTING FLAG
[r,q] = size(p);
[s3,q] = size(t);
plottype = max(r,s3) == 1;

% PLOTTING
newplot;
message = sprintf('TRAINBP: %%g/%g epochs, SSE = %%g.\n',me);
fprintf(message,0,SSE)
if plottype
  h = plotfa(p,t,p,a3);
else
  h = ploterr(tr(1),eg);
end

for i=1:me
```

```
    % CHECK PHASE
    if SSE < eg, i=i-1; break, end

    % BACKPROPAGATION PHASE
    d3 = feval(df3,a3,e);
    d2 = feval(df2,a2,d3,w3);
    d1 = feval(df1,a1,d2,w2);

    % LEARNING PHASE
    [dw1,db1] = learnbp(p,d1,lr);
    [dw2,db2] = learnbp(a1,d2,lr);
    [dw3,db3] = learnbp(a2,d3,lr);
    w1 = w1 - dw1; b1 = b1 + db1;
    w2 = w2 - dw2; b2 = b2 + db2;
    w3 = w3 - dw3; b3 = b3 + db3;

    % PRESENTATION PHASE
    a1 = feval(f1,w1 * p,b1);
    a2 = feval(f2,w2 * a1,b2);
    a3 = feval(f3,w3 * a2,b3);
    e = t-a3;
    SSE = sumsqr(e);

    % TRAINING RECORD
    tr(i+1) = SSE;

    % PLOTTING
    if rem(i,df) == 0
    fprintf(message,i,SSE)
      if plottype
        delete(h);
        h = plot(p,a2);
        drawnow;
      else
        h = ploterr(tr(1:(i+1)),eg,h);
      end
    end
end

% TRAINING RECORD
tr = tr(1:(i+1));

% PLOTTING
if rem(i,df) ~= 0
```

```
  fprintf(message,i,SSE)
  if plottype
    delete(h);
    plot(p,a2);
    drawnow;
  else
    ploterr(tr,eg,h);
  end
end

% WARNINGS
if SSE > eg
  disp('        ')
  disp('TRAINBP: Network error did not reach the error goal.')
  disp('  Further training may be necessary, or try different')
  disp('  initial weights and biases and/or more hidden neurons.')
  disp('     ')
end
```

113. function [w1,b1,i,tr] = tbpx1(w1,b1,f1,p,t,tp)

函数 TBPX1 用于训练一层网络的快速误差反传方法。

【例子】

[W,B,TE,TR] = TBPX1(W,B,F,P,T,TP)

输入参数：

W 为(SxR)阶权重矩阵；

B 为(Sx1)阶的偏置列矢量；

字符串参数 F 为转换(或活性)函数：

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(SxQ)阶的目标矩阵。

TP 为训练参数 (可选择).

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为学习率(Learning rate)，预置值为0.01；

TP(5)为学习率增量，预置值为 1.05；

TP(6)为学习率减量，预置值为 0.7；

TP(7)为 Momentum constant，预置值为 0.9；

TP(8)为最大误差率(Maximum error ratio)，预置值为1.04。

没给训练参数的都将采用预置值。

【程序】

```
if nargin < 5,error('Not enough arguments.');end

% TRAINING PARAMETERS
if nargin == 5, tp = []; end
tp = nndef(tp,[25 1000 0.02 0.01 1.05 0.7 0.9 1.04]);
df = tp(1);
me = tp(2);
eg = tp(3);
lr = tp(4);
im = tp(5);
dm = tp(6);
mc = tp(7);
er = tp(8);
df1 = feval(f1,'delta');

dw1 = w1 * 0;
db1 = b1 * 0;
MC = 0;

% PRESENTATION PHASE
a1 = feval(f1,w1 * p,b1);
e = t-a1;
SSE = sumsqr(e);

% TRAINING RECORD
tr = zeros(2,me+1);
tr(1:2,1) = [SSE; lr];

% PLOTTING FLAG
[r,q] = size(p);
[s1,q] = size(t);
plottype = (max(r,s1) == 1) & 0;

% PLOTTING
newplot;
message = sprintf('TRAINBPX: %%   g/%g epochs, lr = %%   g, SSE = %%   g.\n',me);
fprintf(message,0,lr,SSE)
if plottype
  h = plotfa(p,t,p,a1);
else
  h = plottr(tr(1:2,1),eg);
end
```

```
% BACKPROPAGATION PHASE
d1 = feval(df1,a1,e);

for i=1:me

  % CHECK PHASE
  if SSE < eg, i=i-1; break, end

  % LEARNING PHASE
  [dw1,db1] = learnbpm(p,d1,lr,MC,dw1,db1);
  MC = mc;
  new_w1 = w1 + dw1; new_b1 = b1 + db1;

  % PRESENTATION PHASE
  new_a1 = feval(f1,new_w1 * p,new_b1);
  new_e = t-new_a1;
  new_SSE = sumsqr(new_e);

  % MOMENTUM & ADAPTIVE LEARNING RATE PHASE
  if new_SSE > SSE * er
    lr = lr * dm;
    MC = 0;
  else
    if new_SSE < SSE
      lr = lr * im;
      end
    w1 = new_w1; b1 = new_b1; a1 = new_a1;
    e = new_e; SSE = new_SSE;

    % BACKPROPAGATION PHASE
    d1 = feval(df1,a1,e);
  end

  % TRAINING RECORD
  tr(:,i+1) = [SSE; lr];

  % PLOTTING
  if rem(i,df) == 0
    fprintf(message,i,lr,SSE)
    if plottype
      delete(h);
      h = plot(p,a1);
    else
      h = plottr(tr(1:2,1:(i+1)),eg,h);
```

```
      end
    end
  end

% TRAINING RECORD
tr = tr(1:2,1:(i+1));

% PLOTTING
if rem(i,df) ~= 0
  fprintf(message,i,lr,SSE)
  if plottype
    delete(h).
    plot(p,a1);
  else
    plottr(tr,eg,h);
  end
end

% WARNINGS
if SSE > eg
  disp('    ')
  disp('TRAINBPX: Network error did not reach the error goal.')
  disp('  Further training may be necessary, or try different')
  disp('  initial weights and biases and/or more layers.')
  disp('    ')
end
```

114. function [w1,b1,w2,b2,i,tr] = tbpx2(w1,b1,f1,w2,b2,f2,p,t,tp)

函数 TBPX2 用于训练二层网络的快速误差反传方法。

【例子】

% [W1,B1,W2,B2,TE,TR] = TBPX2(W2,B2,F1,W1,B1,F2,P,T,TP)

输入参数：

Wi 为第 i 层权重矩阵；

Bi 为第 i 层偏置列矢量；

字符串参数 Fi 为转换(或活性)函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(SxQ)阶的目标矩阵；

TP 为训练参数（可选择)。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为学习率(Learning rate)，预置值为0.01；

TP(5)为学习率增量，预置值为 1.05；

TP(6)为学习率减量，预置值为 0.7；

TP(7)为 Momentum constant，预置值为 0.9；

TP(8)为最大误差率(Maximum error ratio)，预置值为1.04。

没给训练参数的都将采用预置值。

【程序】

```
if nargin < 8,error('Not enough arguments.');end

% TRAINING PARAMETERS
if nargin == 8, tp = []; end
tp = nndef(tp,[25 1000 0.02 0.01 1.05 0.7 0.9 1.04]);
df = tp(1);
me = tp(2);
eg = tp(3);
lr = tp(4);
im = tp(5);
dm = tp(6);
mc = tp(7);
er = tp(8);
df1 = feval(f1,'delta');
df2 = feval(f2,'delta');

dw1 = w1 * 0;
db1 = b1 * 0;
dw2 = w2 * 0;
db2 = b2 * 0;
MC = 0;

% PRESENTATION PHASE
a1 = feval(f1,w1 * p,b1);
a2 = feval(f2,w2 * a1,b2);
e = t-a2;
SSE = sumsqr(e);

% TRAINING RECORD
tr = zeros(2,me+1);
tr(1:2,1) = [SSE; lr];

% PLOTTING FLAG
[r,q] = size(p);
[s,q] = size(t);
```

```
plottype = (max(r,s) == 1);

% PLOTTING
newplot;
message = sprintf('TRAINBPX: %%g/%g epochs, lr = %%g, SSE = %%g.\n',me);
fprintf(message,0,lr,SSE)
if plottype
  h = plotfa(p,t,p,a2);
else
  h = plottr(tr(1:2,1),eg);
end

% BACKPROPAGATION PHASE
d2 = feval(df2,a2,e);
d1 = feval(df1,a1,d2,w2);

for i=1:me

  % CHECK PHASE
  if SSE < eg, i=i-1; break, end

  % LEARNING PHASE
  [dw1,db1] = learnbpm(p,d1,lr,MC,dw1,db1);
  [dw2,db2] = learnbpm(a1,d2,lr,MC,dw2,db2);
  MC = mc;
  new_w1 = w1 + dw1; new_b1 = b1 + db1;
  new_w2 = w2 + dw2; new_b2 = b2 + db2;

  % PRESENTATION PHASE
  new_a1 = feval(f1,new_w1*p,new_b1);
  new_a2 = feval(f2,new_w2*new_a1,new_b2);
  new_e = t-new_a2;
  new_SSE = sumsqr(new_e);

  % MOMENTUM & ADAPTIVE LEARNING RATE PHASE
  if new_SSE > SSE*er
    lr = lr * dm;
    MC = 0;
  else
    if new_SSE < SSE
      lr = lr * im;
      end
    w1 = new_w1; b1 = new_b1; a1 = new_a1;
    w2 = new_w2; b2 = new_b2; a2 = new_a2;
```

```
    e = new _e; SSE = new _ SSE;

    % BACKPROPAGATION PHASE
    d2 = feval(df2,a2,e);
    d1 = feval(df1,a1,d2,w2);
  end
  % TRAINING RECORD
  tr(:,i+1) = [SSE; lr];
  % PLOTTING
  if rem(i,df) == 0
    fprintf(message,i,lr,SSE)
    if plottype
      delete(h);
      h = plot(p,a2);
      drawnow
    else
      h = plottr(tr(1:2,1:(i+1)),eg,h);
    end
  end
end
% TRAINING RECORD
tr = tr(1:2,1:(i+1));
% PLOTTING
if rem(i,df) ~= 0
  fprintf(message,i,lr,SSE)
  if plottype
    delete(h);
    plot(p,a2);
    drawnow
  else
    plottr(tr,eg,h);
  end
end
% WARNINGS
if SSE > eg
  disp('    ')
  disp('TRAINBPX: Network error did not reach the error goal.')
  disp('  Further training may be necessary, or try different')
  disp('  initial weights and biases and/or more hidden neurons.')
  disp('    ')
end
```

115. function [w1,b1,w2,b2,w3,b3,i,tr]=tbpx3(w1,b1,f1,w2,b2,f2,w3,b3,f3,p,t,tp)

函数 TBPX3 用于训练三层网络的快速误差反传方法。

【例子】

[W1,B1,W2,B2,W3,B3,TE,TR] = TBPX3(W1,B2,F1,W1,B1,F2,W3,B3,F3,P,T,TP)

输入参数：

Wi 为第 i 层权重矩阵；

Bi 为第 i 层偏置列矢量；

字符串参数 Fi 为转换(或活性)函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(SxQ)阶的目标矩阵；

TP 为训练参数（可选择）。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为学习率(Learning rate)，预置值为0.01；

TP(5)为学习率增量，预置值为 1.05；

TP(6)为学习率减量，预置值为 0.7；

TP(7)为 Momentum constant，预置值为 0.9；

TP(8)为最大误差率(Maximum error ratio)，预置值为1.04。

没给训练参数的都将采用预置值。

【程序】

```
if nargin < 11,error('Not enough arguments.');end

% TRAINING PARAMETERS
if nargin == 11, tp = []; end
tp = nndef(tp,[25 1000 0.02 0.01 1.05 0.7 0.9 1.04]);
df = tp(1);
me = tp(2);
eg = tp(3);
lr = tp(4);
im = tp(5);
dm = tp(6);
mc = tp(7);
er = tp(8);
df1 = feval(f1,'delta');
df2 = feval(f2,'delta');
df3 = feval(f3,'delta');

dw1 = w1 * 0;
db1 = b1 * 0;
```

```
dw2 = w2 * 0;
db2 = b2 * 0;
dw3 = w3 * 0;
db3 = b3 * 0;
MC = 0;

% PRESENTATION PHASE
a1 = feval(f1,w1 * p,b1);
a2 = feval(f2,w2 * a1,b2);
a3 = feval(f3,w3 * a2,b3);
e = t-a3;
SSE = sumsqr(e);

% TRAINING RECORD
tr = zeros(2,me+1);
tr(1:2,1) = [SSE; lr];

% PLOTTING FLAG
[r,q] = size(p);
[s,q] = size(t);
plottype = (max(r,s) == 1) & 0;

% PLOTTING
newplot;
message = sprintf('TRAINBPX: %%g/%g epochs, lr = %%g, SSE = %%g.\n',me);
fprintf(message,0,lr,SSE)
if plottype
  h = plotfa(p,t,p,a3);
else
  h = plottr(tr(1:2,1),eg);
end

% BACKPROPAGATION PHASE
d3 = feval(df3,a3,e);
d2 = feval(df2,a2,d3,w3);
d1 = feval(df1,a1,d2,w2);

for i=1:me

  % CHECK PHASE
  if SSE < eg, i=i-1; break, end

  % LEARNING PHASE
  [dw1,db1] = learnbpm(p,d1,lr,MC,dw1,db1);
```

```
[dw2,db2] = learnbpm(a1,d2,lr,MC,dw2,db2);
[dw3,db3] = learnbpm(a2,d3,lr,MC,dw3,db3);
MC = mc;
new_w1 = w1 + dw1; new_b1 = b1 + db1;
new_w2 = w2 + dw2; new_b2 = b2 + db2;
new_w3 = w3 + dw3; new_b3 = b3 + db3;

% PRESENTATION PHASE
new_a1 = feval(f1,new_w1 * p,new_b1);
new_a2 = feval(f2,new_w2 * new_a1,new_b2);
new_a3 = feval(f3,new_w3 * new_a2,new_b3);
new_e = t - new_a3;
new_SSE = sumsqr(new_e);

% MOMENTUM & ADAPTIVE LEARNING RATE PHASE
if new_SSE > SSE * er
  lr = lr * lm;
  MC = 0;
else
  if new_SSE < SSE
    lr = lr * im;
    end
  w1 = new_w1; b1 = new_b1; a1 = new_a1;
  w2 = new_w2; b2 = new_b2; a2 = new_a2;
  w3 = new_w3; b3 = new_b3; a3 = new_a3;
  e = new_e; SSE = new_SSE;

  % BACKPROPAGATION PHASE
  d3 = feval(df3,a3,e);
  d2 = feval(df2,a2,d3,w3);
  d1 = feval(df1,a1,d2,w2);
end

% TRAINING RECORD
tr(1:2,i+1) = [SSE; lr];

% PLOTTING
if rem(i,df) == 0
  fprintf(message,i,lr,SSE)
  if plottype
    delete(h);
    h = plot(p,a3);
  else
    h = plottr(tr(1:2,1:(i+1)),eg,h);
```

```
      end
    end
end

% TRAINING RECORD
tr = tr(1:2,1:(i+1));

% PLOTTING
if rem(i,df) ~= 0
  fprintf(message,i,lr,SSE)
  if plottype
    delete(h);
    plot(p,a3);
  else
    plottr(tr,eg,h);
  end
end

% WARNINGS
if SSE > eg
  disp('    ')
  disp('TRAINBPX: Network error did not reach the error goal.')
  disp('  Further training may be necessary, or try different')
  disp('  initial weights and biases and/or more hidden neurons.')
  disp('    ')
end
```

116. function [w1,b1,i,tr] = tlm1(w1,b1,f1,p,t,tp)

函数 TLM1 训练一层前传误差反传网络的 Levenberg-Marquardt 方法.

【例子】

[W,B,TE,TR] = TLM1(W,B,'F1',P,T)

输入参数:

W 为权重矩阵;

B 为偏置列矢量;

字符串参数 F 为转换函数;

P 为包含输入矢量的(RxQ)阶的输入矩阵;

T 为包含目标矢量的(S1xQ)阶的目标矩阵;

TP 为训练参数（可选择)。

输出参数:

Wi 为新权重;

Bi 为新偏置;

TE 为实际的训练数目;

TR 为训练记录(误差)。

可供选择的训练参数:

TP(1)为显示之间的训练数目,预置值为25;

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为最小梯度，预置值为0.0001；

TP(5)为 MU 的初始值，预置值为0.001；

TP(6)为 MU 增量的乘数，预置值为 10；

TP(7)为 MU 减量的乘数，预置值为 0.1；

TP(8)为 MU 的最大值，预置值为1^{10}。

没给训练参数的都将采用预置值。

【程序】

```
if nargin < 5,error('Not enough arguments.'),end

% TRAINING PARAMETERS
if nargin == 5, tp = []; end
tp = nndef(tp,[25 1000 0.02 0.0001 0.001 10 0.1 1e10]);
df = tp(1);
me = tp(2);
eg = tp(3);
grad_min = tp(4);
mu_init = tp(5);
mu_inc = tp(6);
mu_dec = tp(7);
mu_max = tp(8);
df1 = feval(f1,'delta');

% DEFINE SIZES
[s1,r] = size(w1);
w1_ind = [1:(s1*r)];
b1_ind = [1:s1] + w1_ind(length(w1_ind));
ii = eye(b1_ind(length(b1_ind)));
dw1 = w1; db1 = b1;
ext_p = nncpyi(p,s1);

% PRESENTATION PHASE
a1 = simuff(p,w1,b1,f1);
e = t-a1;
SSE = sumsqr(e);

% TRAINING RECORD
tr = zeros(1,me+1);
tr(1) = SSE;

% PLOTTING FLAG
plottype = (r==1) & (s1==1);
```

```
% PLOTTING
newplot;
message = sprintf('TRAINLM: %%g/%g epochs, mu = %%g, SSE = %%g.\n',me);
fprintf(message,0,mu_init,SSE)
if plottype
  h = plotfa(p,t,p,a1);
else
  h = ploterr(tr(1),eg);
end

mu = mu_init;
for i=1:me

   % CHECK PHASE
   if SSE < eg, i=i-1; break, end

   % FIND JACOBIAN
   d1 = feval(df1,a1);
   ext_d1 = -nncpyd(d1);
   j1 = learnlm(ext_p,ext_d1);
                  j = [j1, ext_d1'];

   % CHECK MAGNITUDE OF GRADIENT
   je = j' * e(:);
                  grad = norm(je);
                  if grad < grad_min, i=i-1; break, end

   % INNER LOOP, INCREASE MU UNTIL THE ERRORS ARE REDUCED
   jj = j' * ;

                  while (mu <= mu_max)
                        dx = -(jj+ii * mu) \ je;
     dw1(:) = dx(w1_ind); db1 = dx(b1_ind);
                              new_w1 = w1 + dw1; new_b1 = b1 + db1;

     % EVALUATE NEW NETWORK
     a1 = simuff(p,new_w1,new_b1,f1);
                        new_e = t-a1;
                        new_SSE = sumsqr(new_e);

     if (new_SSE < SSE), break, end
     mu = mu * mu_inc;
                  end
   if (mu > mu_max), i = i-1; break, end
```

```
            mu = mu * mu_dec;

  % UPDATE NETWORK
            w1 = new_w1; b1 = new_b1;
  e = new_e; SSE = new_SSE;

  % TRAINING RECORD
  tr(i+1) = SSE;

  % PLOTTING
  if rem(i,df) == 0
    fprintf(message,i,mu,SSE)
    if plottype
      delete(h); h = plot(p,a1,'m'); drawnow;
    else
      h = ploterr(tr(1:(i+1)),eg,b);
    end
  end
end

% TRAINING RECORD
tr = tr(1:(i+1));

% PLOTTING
if rem(i,df) ~= 0
  fprintf(message,i,mu,SSE)
  if plottype
    delete(h);
    plot(p,a1,'m');
    drawnow;
  else
    ploterr(tr,eg,h);
  end
end

% WARNINGS
if SSE > eg
  disp(' ')
  if (mu > mu_max)
    disp('TRAINLM: Error gradient is too small to continue learning.')
  else
    disp('TRAINLM: Network error did not reach the error goal.')
  end
  disp(' Further training may be necessary, or try different  ')
```

```
  disp(' initial weights and biases and/or more hidden layers.')
  disp('          ')
end
```

117. function [w1,b1,w2,b2,i,tr] = tlm2(w1,b1,f1,w2,b2,f2,p,t,tp)

函数 TLM2 采用 Levenberg-Marquardt 方法，用于训练二层前传误差反传网络。

【例子】

[W1,B1,W2,B2,TE,TR] = TLM2(W1,B1,'F1',W2,B2,'F2',P,T)

输入参数：

Wi 为第 i 层权重矩阵；

Bi 为第 i 层偏置列矢量；

字符串参数 Fi 为转换(或活性)函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(S2xQ)阶的目标矩阵；

TP 为训练参数（可选择）。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为最小梯度，预置值为1^{-6}；

TP(5)为 MU 的初始值，预置值为0.001；

TP(6)为 MU 增量的乘数，预置值为 10；

TP(7)为 MU 减量的乘数，预置值为 0.1；

TP(8)为 MU 的最大值，预置值为1^{10}。

没给训练参数的都将采用预置值。

【程序】

```
if nargin < 8,error('Not enough arguments.'),end

% TRAINING PARAMETERS
if nargin == 8, tp = []; end
tp = nndef(tp,[25 1000 0.02 1e-6 0.001 10 0.1 1e10]);
df = tp(1);
me = tp(2);
eg = tp(3);
grad_min = tp(4);
mu_init = tp(5);
mu_inc = tp(6);
mu_dec = tp(7);
mu_max = tp(8);
df1 = feval(f1,'delta');
```

```
df2 = feval(f2,'delta');

% DEFINE SIZES
[s1,r] = size(w1);
[s2,s1] = size(w2);
w1_ind = [1:(s1 * r)];
b1_ind = [1:s1] + w1_ind(length(w1_ind));
w2_ind = [1:(s1 * s2)] + b1_ind(length(b1_ind));
b2_ind = [1:s2] + w2_ind(length(w2_ind));
ii = eye(b2_ind(length(b2_ind)));
dw1 = w1; db1 = b1;
dw2 = w2; db2 = b2;
ext_p = nncpyi(p,s2);

% PRESENTATION PHASE
[a1,a2] = simuff(p,w1,b1,f1,w2,b2,f2);
e = t-a2;
SSE = sumsqr(e);

% TRAINING RECORD
tr = zeros(1,me+1);
tr(1) = SSE;

% PLOTTING FLAG
plottype = (r==1) & (s2==1);

% PLOTTING
newplot;
message = sprintf('TRAINLM: %%g/%g epochs, mu = %%g, SSE = %%g.\n',me);
fprintf(message,0,mu_init,SSE)
if plottype
  h = plotfa(p,t,p,a2);
else
  h = ploterr(tr(1),eg);
end

mu = mu_init;
for i=1:me

  % CHECK PHASE
  if SSE < eg, i=i-1; break, end
  % FIND JACOBIAN

  ext_a1 = nncpyi(a1,s2);
```

```
d2 = feval(df2,a2);
ext_d2 = -nncpyd(d2);
ext_d1 = feval(df1,ext_a1,ext_d2,w2);
j1 = learnlm(ext_p,ext_d1);
j2 = learnlm(ext_a1,ext_d2);
j = [j1, ext_d1', j2, ext_d2'];

% CHECK MAGNITUDE OF GRADIENT
je = j' * e(:);
grad = norm(je);
if grad < grad_min, i=i-1; break, end

% INNER LOOP, INCREASE MU UNTIL THE ERRORS ARE REDUCED
jj = j' * j;

while (mu <= mu_max)
  dx = -(jj+ii * mu) \ je;
  dw1(:) = dx(w1_ind); db1 = dx(b1_ind);
  dw2(:) = dx(w2_ind); db2 = dx(b2_ind);
  new_w1 = w1 + dw1; new_b1 = b1 + db1;
  new_w2 = w2 + dw2; new_b2 = b2 + db2;

% EVALUATE NEW NETWORK
  [a1,a2] = simuff(p,new_w1,new_b1,f1,new_w2,new_b2,f2);
  new_e = t-a2;
  new_SSE = sumsqr(new_e);

  if (new_SSE < SSE), break, end
  mu = mu * mu_inc;
end
if (mu > mu_max), i = i-1; break, end
mu = mu * mu_dec;

% UPDATE NETWORK
w1 = new_w1; b1 = new_b1;
w2 = new_w2; b2 = new_b2;
e = new_e; SSE = new_SSE;

% TRAINING RECORD
tr(i+1) = SSE;

% PLOTTING
if rem(i,df) == 0
  fprintf(message,i,mu,SSE)
```

```
    if plottype
      delete(h); h = plot(p,a2,'  m  '); drawnow;
    else
      h = ploterr(tr(1:(i+1)),eg,h);
    end
  end
end

% TRAINING RECORD
tr = tr(1:(i+1));

% PLOTTING
if rem(i,df) ~= 0
  fprintf(message,i,mu,SSE)
  if plottype
    delete(h);
    plot(p,a2,'  m  ');
    drawnow;
  else
    ploterr(tr,eg,h);
  end
end

% WARNINGS
if SSE > eg
  disp('    ')
  if (mu > mu_max)
    disp('TRAINLM: Error gradient is too small to continue learning.')
  else
    disp('TRAINLM: Network error did not reach the error goal.')
  end
  disp('  Further training may be necessary, or try different  ')
  disp('  initial weights and biases and/or more hidden neurons.')
  disp('    ')
end
```

118. function [w1,b1,w2,b2,w3,b3,i,tr] = tlm3(w1,b1,f1,w2,b2,f2,w3,b3,f3,p,t,tp)

函数 TLM3 采用 Levenberg-Marquardt 方法，用于训练三层前向误差反传网络。

【例子】

[W1,B1,W2,B2,W3,B3,TE,TR] = TLM3(W1,B1,F1,W2,B2,F2,W3,B3,F3,P,T)

输入参数：

Wi 为第 i 层权重矩阵；

Bi 为第 i 层偏置列矢量；

字符串参数 Fi 为转换(或活性)函数；

P 为包含输入矢量的(RxQ)阶的输入矩阵；

T 为包含目标矢量的(S2xQ)阶的目标矩阵；

TP 为训练参数（可选择）。

输出参数：

Wi 为新权重；

Bi 为新偏置；

TE 为实际的训练数目；

TR 为训练记录(误差)。

可供选择的训练参数：

TP(1)为显示之间的训练数目，预置值为25；

TP(2)为训练的最大数目，预置值为1000；

TP(3)为平方误差和目标，预置值为 0.02；

TP(4)为最小梯度，预置值为0.0001；

TP(5)为 MU 的初始值，预置值为0.001；

TP(6)为 MU 增量的乘数，预置值为 10；

TP(7)为 MU 减量的乘数，预置值为 0.1；

TP(8)为 MU 的最大值，预置值为1^{10}。

没给训练参数的都将采用预置值。

【程序】

```
if nargin < 11,error('Not enough arguments.'),end

% TRAINING PARAMETERS
if nargin == 11, tp = []; end
tp = nndef(tp,[25 1000 0.02 0.0001 0.001 10 0.1 1e10]);
df = tp(1);
me = tp(2);
eg = tp(3);
grad_min = tp(4);
mu_init = tp(5);
mu_inc = tp(6);
mu_dec = tp(7);
mu_max = tp(8);
df1 = feval(f1,'delta');
df2 = feval(f2,'delta');
df3 = feval(f3,'delta');

% DEFINE SIZES
[s1,r] = size(w1);
[s2,s1] = size(w2);
[s3,s2] = size(w3);
w1_ind = [1:(s1*r)];
b1_ind = [1:s1] + w1_ind(length(w1_ind));
w2_ind = [1:(s1*s2)] + b1_ind(length(b1_ind));
b2_ind = [1:s2] + w2_ind(length(w2_ind));
w3_ind = [1:(s2*s3)] + b2_ind(length(b2_ind));
```

```
b3_ind = [1:s3] + w3_ind(length(w3_ind));
ii = eye(b3_ind(length(b3_ind)));
dw1 = w1; db1 = b1;
dw2 = w2; db2 = b2;
dw3 = w3; db3 = b3;
ext_p = nncpyi(p,s3);

% PRESENTATION PHASE
[a1,a2,a3] = simuff(p,w1,b1,f1,w2,b2,f2,w3,b3,f3);
e = t-a3;
SSE = sumsqr(e);

% TRAINING RECORD
tr = zeros(1,me+1);
tr(1) = SSE;

% PLOTTING FLAG
plottype = (r==1) & (s2==1);

% PLOTTING
newplot;
message = sprintf('TRAINLM: %%g/%g epochs, mu = %%g, SSE = %%g.\n',me);
fprintf(message,0,mu_init,SSE)
if plottype
  h = plotfa(p,t,p,a3);
else
  h = ploterr(tr(1),eg);
end
mu = mu_init;
for i=1:me

  % CHECK PHASE
  if SSE < eg, i=i-1; break, end

  % FIND JACOBIAN
                ext_a1 = nncpyi(a1,s3);
                ext_a2 = nncpyi(a2,s3);
  d3 = feval(df3,a3);
  ext_d3 = -nncpyd(d3);
  ext_d2 = feval(df2,ext_a2,ext_d3,w3);
  ext_d1 = feval(df1,ext_a1,ext_d2,w2);
  j1 = learnlm(ext_p,ext_d1);
  j2 = learnlm(ext_a1,ext_d2);
  j3 = learnlm(ext_a2,ext_d3);
```

```
j = [j1, ext_d1', j2, ext_d2', j3, ext_d3'];

% CHECK MAGNITUDE OF GRADIENT
je = j' * e(:);
            grad = norm(je);
            if grad < grad_min, i=i-1; break, end

% INNER LOOP, INCREASE MU UNTIL THE ERRORS ARE REDUCED
jj = j' * j;

            while (mu <= mu_max)
                  dx = -(jj+ii * mu) \ je;
  dw1(:) = dx(w1_ind); db1 = dx(b1_ind);
  dw2(:) = dx(w2_ind); db2 = dx(b2_ind);
  dw3(:) = dx(w3_ind); db3 = dx(b3_ind);
  new_w1 = w1 + dw1; new_b1 = b1 + db1;
  new_w2 = w2 + dw2; new_b2 = b2 + db2;
  new_w3 = w3 + dw3; new_b3 = b3 + db3;

  % EVALUATE NEW NETWORK
  [a1,a2,a3] =...
simuff(p,new_w1,new_b1,f1,new_w2,new_b2,f2,new_w3,new_b3,f3);
                  new_e = t-a3;
                  new_SSE = sumsqr(new_e);

  if (new_SSE < SSE), break, end
  mu = mu * mu_inc;
            end
 if (mu > mu_max), i = i-1; break, end
            mu = mu * mu_dec;

 % UPDATE NETWORK
 w1 = new_w1; b1 = new_b1;
 w2 = new_w2; b2 = new_b2;
 w3 = new_w3; b3 = new_b3;
 e = new_e; SSE = new_SSE;

 % TRAINING RECORD
 tr(i+1) = SSE;

 % PLOTTING
 if rem(i,df) == 0
   fprintf(message,i,mu,SSE)
```

```
      if plottype
        delete(h); h = plot(p,a2,'m'); drawnow;
      else
        h = ploterr(tr(1:(i+1)),eg,h);
      end
    end
  end

% TRAINING RECORD
tr = tr(1:(i+1));

% PLOTTING
if rem(i,df) ~= 0
  fprintf(message,i,mu,SSE)
  if plottype
    delete(h);
    plot(p,a2,'m');
    drawnow;
  else
    ploterr(tr,eg,h);
  end
end

% WARNINGS
if SSE > eg
  disp('      ')
  if (mu > mu_max)
    disp('TRAINLM: Error gradient is too small to continue learning.')
  else
    disp('TRAINLM: Network error did not reach the error goal.')
  end
  disp(' Further training may be necessary, or try different')
  disp(' initial weights and biases and/or more hidden neurons.')
  disp('        ')
end
```

119. function lmsobj=lmsobj(c,x,y);

函数 lmsobj 为最小方差中位数目标函数子程序，为函数 sntopti 调用子程序。

【例子】

lmsobj=lmsobj(c,x,y);

输入参数：c 为浓度矢量；

　　　　x 为校正矩阵；

　　　　y 为响应矢量；

输出参数：lmsobj 所得目标函数值。

【程序】

```
error=x*c-y;
lmsobj=median(error.*error);
```

第十五章　MATLAB 原程序变量说明与使用示例

一、MATLAB 中最优化计算工具箱（optimization toolbox）使用示例

1. 考虑以下函数的无约束优化（极小值）问题

$$f(x) = \exp(x(1)).(4x(1)^2 + 2x(2)^2 + 4x(1).x(2) + 2x(2) + 1)$$

(1) 定义目标函数

```
》fun = 'exp(x(1)) * (4*x(1)^2 + 2*x(2)^2 + 4*x(1)*x(2) + 2*x(2) + 1)';
```

(2) 给出初始搜寻点

```
》x0 = [-1 1];
```

(3) 用预置的参数值

```
》options = [];
```

(4) 采用函数 fminu 对此问题进行优化

```
》[x, options] = fminu(fun, x0, options);
```

(5) 其最优坐标位置为

```
》x
x =
   0.5000 -1.0000
```

(6) 其最优解为

```
》options(8)
ans =
  1.3029e-010
```

(7) 该优化过程对其进行计算的总次数为

```
》options(10)
ans =
      36
```

2. 考虑以下函数的有约束优化（极小值）问题

$$\text{minimize}\quad f(x) = \exp(x(1)).(4x(1)^2 + 2x(2)^2 + 4x(1).x(2) + 2x(2) + 1)$$

约束条件：

$$1.5 + x(1).x(2) - x(1) - x(2) <= 0$$

$$-x(1).x(2) <= 10$$

(1) 定义目标函数和约束条件

```
》funf = 'f = exp(x(1)) * (4*x(1)^2 + 2*x(2)^2 + 4*x(1)*x(2) + 2*x(2) + 1);';
》fung = 'g = [1.5 + x(1)*x(2) - x(1) - x(2); -x(1)*x(2) - 10];';
》fun = [funf fung];
```

(2) 给出初始搜寻点

```
》x0 = [-1 1];
```

(3) 用预置的参数值

```
》options = [];
```

(4) 采用函数 constr 对此问题进行优化

```
》[x, options] = constr(fun, x0, options);
```

(5) 其最优坐标位置为

```
› x
x =
    -9.5474   1.0474
```

(6) 其最优解为

```
› options(8)
ans =
    0.0236
```

(7) 其解的约束条件为

```
› g = [1.5 + x(1) * x(2) - x(1) - x(2); - x(1) * x(2) - 10]
g =
  1.0e-015 *
  -0.8882
        )
```

(8) 该优化过程对其进行计算的总次数为

```
› options(10)
ans =
    29
```

3. 考虑以下函数的有函数和边界约束的优化（极小值）问题

minimize f(x) = exp(x(1)) . (4x(1)^2+ 2x(2)^2+ 4x(1). x(2) + 2x(2) -- 1)

约束条件：　　1.5 + x(1). x(2) − x(1) − x(2) <= 0

　　− x(1). x (2) <= 10

及　　x(1) >= 0

　　x(2) >= 0

(1) 定义目标函数和约束条件

```
› funf = 'f = exp(x(1)) * (4 * x(1)^ 2 + 2 * x(2)^ 2 + 4 * x(1) * x(2) + 2 * x(2) + 1);';
› fung = 'g = [1.5 + x(1) * x(2) - x(1) - x(2); -x(1) * x(2) - 10];';
› fun = [funf fung];
```

(2) 给出初始搜寻点

```
› x0 = [-1 1];
```

(3) 用预置的参数值

```
› options = [ ];
```

(4) 用函数 constr 来进行优化计算

```
› [x, options] = constr(fun, x0, options, vlb, vub);
```

在此输入参数 vlb 和 vub 为

```
› vlb = zeros(1,2);   % 下界 X >= 0
› vub = [ ];          %  没有上界
```

(5) 其最优坐标位置为

```
› x
x =
    0     1.5000
```

(6) 其最优解为

```
› options(8)
ans =
```

```
   8.5000
```

(7) 其解的约束条件为

```
 》g = [1.5 - x(1)*x(2) - x(1) - x(2); - x(1)*x(2) - 10]
g =
     0
   -10
```

(8) 该优化过程对其进行计算的总次数为

```
 》options(10)
ans =
    10
```

4. 如果由使用者提供导函数，可加快运算，仍考虑以下函数的有约束的优化问题

minimize $f(x) = \exp(x(1)) \cdot (4x(1)^2 + 2x(2)^2 - 4x(1) \cdot x(2) + 2x(2) + 1)$

约束条件：　$1.5 + x(1) \cdot x(2) - x(1) - x(2) <= 0$

$x(1) \cdot x(2) <= 10$

(1) 定义目标函数和约束条件

```
》funf = 'f = exp(x(1)) * (4 * x(1)^2 - 2 * x(2)^2 + 4 * x(1) * x(2) + 2 * x(2) + 1);';
》fung = 'g = [1.5 - x(1) * x(2) - x(1)    x(2); -x(1) * x(2) -- 10];';
》fun = [funf fung];
```

(2) 给出初始搜寻点

```
》x0 = [-1 1];
```

(3) 用预置的参数值

```
》options = [ ];
```

(4) 定义函数和约束条件的导函数矩阵

```
》dfdx1 = 'exp(x(1)) * (4 * x(1)^2+2 * x(2)^2-4 * x(1) * x(2)+2 * x(2)+1)+4 * exp(x(1)) * (2 * x
(1)+x(2));';
》dfdx2 = '4 * exp(x(1)) * (x(1)+x(2)+0.5);';
》dfdg = '[x(2)-1, -x(2); x(1)-1, -x(1)];';
》grad = ['df=[',dfdx1, dfdx2,']; dg=', dfdg];
```

(5) 用函数 constr 来进行优化计算：

```
》[x, options] = constr(fun, x0, options, [ ], [ ], grad);
```

(6) 其最优坐标位置为

```
》x
x =
  -9.5474     1.0474
```

(7) 其最优解为

```
》options(8)
ans =
    0.0236
```

(8) 其解的约束条件为

```
》g = [1.5 + x(1) * x(2) - x(1) - x(2); - x(1) * x(2) - 10]
g =
  1.0e-014 *
      0.1110
     -0.1776
```

(9) 该优化过程对其进行计算的总次数为

```
》options(10)
ans =
     11
```

5. 由使用者给出不同的选择，可改变优化过程的最终条件和显示模式，仍考虑以下函数的无约束的优化问题：

$$f(x) = \exp(x(1)) \cdot (4x(1)^2 + 2x(2)^2 + 4x(1) \cdot x(2) + 2x(2) + 1)$$

(1) 定义目标函数

```
》fun = 'exp(x(1)) * (4 * x(1)^2 + 2 * x(2)^2 + 4 * x(1) * x(2) + 2 * x(2) - 1)';
```

(2) 给出初始搜寻点

```
》x0 = [-1 1];
```

(3) 重新给出终止标准

```
》options(2) = 1e-8;     % 坐标点终止标准；
》options(3) = 1e-8;     % 目标函数值终止标准；
```

(4) 用函数 fminu 来进行优化计算

```
》[x, options] = fminu(fun, x0, options);
```

(5) 其最优坐标位置为

```
》x
x =
    0.5000    -1.0000
```

(6) 其最优解为

```
》options(8)
ans =
    3.4412e-014
```

(7) 该优化过程对其进行计算的总次数为

```
》options(10)
ans =
     61
```

(8) 如果想看到该优化过程的迭代情况，则可令

```
》options (1)= 1; % 即表示将选择项中的第一项改为 1。即表示要选择屏幕显示；
```

(9) 用函数 fminu 来进行寻优计算

```
》[x, options] = fminu(fun, x0, options);
```

可得：

f-COUNT	FUNCTION	STEP-SIZE	GRAD/SD
4	1.8394	1	-0.677
9	1.72427	0.361834	-0.00354
17	0.362233	5.90886	-0.295
20	0.221775	3	0.436
25	0.0171251	0.529597	0.00204
30	6.31008e-005	0.25069	-2.92e-005
36	6.21504e-008	1.37782	1.27e-009

Optimization Terminated Successfully

NUMBER OF FUNCTION EVALUATIONS=36

6. 用优化程序来进行曲线拟合示例

(1) 先给出一个数据例子

```
〉Data = ···
[0.0000    5.8955
 0.1000    3.5639
 0.2000    2.5173
 0.3000    1.9790
 0.4000    1.8990
 0.5000    1.3938
 0.6000    1.1359
 0.7000    1.0096
 0.8000    1.0343
 0.9000    0.8435
 1.0000    0.6856
 1.1000    0.6100
 1.2000    0.5392
 1.3000    0.3946
 1.4000    0.3903
 1.5000    0.5474
 1.6000    0.3459
 1.7000    0.1370
 1.8000    0.2211
 1.9000    0.1704
 2.0000    0.2636];
```

(2) 将此数据在屏幕上显示出来

```
〉t = Data(:,1);
〉y = Data(:,2);
〉axis([0 2 -0.5 6])
〉figure(fig)
〉hold on
〉plot(t,y,'ro','EraseMode','none');
〉title('Input data')
```

(3) 用MATLAB的M文件来构造一个函数来拟合上述数据

```
function f = fitfun (lam,Data,Plothandle)
%   FITFUN (lam,Data,Plothandle) returns the error
%   between the data and the values computed by the current
%   function of lam. FITFUN assumes a function of the form
%      y = c(1) * exp(-lam(1) * t) + ··· + c(n) * exp(-lam(n) * t)
%   with n linear parameters and n nonlinear parameters;
t = Data(:,1); y = Data(:,2);
A = zeros(length(t),length(lam));
for j = 1:size(lam)
   A(:,j) = exp(-lam(j) * t);
end
c = A\y;
```

```
z = A * c;
f = z-y;
% Statements to plot progress of fitting:
set(Plothandle,'ydata',z)
drawnow
```

将该函数以 M 文件形式保存下来。

(4) 给出该函数的初始值及原图的句柄

```
> lam = [1 0]';
> h = plot(t,y,'b-','EraseMode','xor');
```

(5) 调用优化程序来进行曲线拟合

在 matlab 的优化工具箱中，可采用下述方法来进行优化：

1) Broyden-Fletcher-Golfarb-Shanno
2) Davidon-Fletcher-Powell
3) Steepest Descent
4) Simplex Search
5) Gauss-Newton
6) Levenberg-Marquardt

如采用方法 1) 来进行上述数据拟合，则有：

```
>OPTIONS=0;
>[lam,OPTIONS]=fminu('norm(fitfun2(x,P1,P2))',lam,OPTIONS,[],Data,h);
```

如采用方法 2) 来进行上述数据拟合，则有：

```
>OPTIONS(6)=1;
>[lam,OPTIONS]=fminu('norm(fitfun2(x,P1,P2))',lam,OPTIONS,[],Data,h);
```

如采用方法 3) 来进行上述数据拟合，则有：

```
>OPTIONS(6)=2;
>[lam,OPTIONS]=fminu('norm(fitfun2(x,P1,P2))',lam,OPTIONS,[],Data,h);
```

如采用方法 4) 来进行上述数据拟合，则有：

```
>OPTIONS(5)=1;
>[lam,OPTIONS]=fmins('norm(fitfun2(x,P1,P2))',lam,OPTIONS,[],Data,h);
```

如采用方法 5)来进行上述数据拟合，则有：

```
>OPTIONS(5)=1;
>[lam,OPTIONS]=leastsq('fitfun2', lam,OPTIONS,[],Data,h);
```

如采用方法 6)来进行上述数据拟合，则有：

```
>OPTIONS=0;
>[lam,OPTIONS]=leastsq('fitfun2', lam,OPTIONS,[],Data,h);
```

二、MATLAB 中最优化计算工具箱通用选择参数（options）说明

最优化计算工具箱通用选择参数（OPTIONS）可用于很多 MATLAB 子程序，如：在 MATLAB 自身中的：FMIN 函数和 FMINS 函数，以及在其最优化计算工具箱中的下列函数中：FMINU，CONSTR，ATTGOAL，MINIMAX，LEASTSQ，FSOLVE.

其预置参数值为：

```
default_options=[0,1e-4,1e-4,1e-6,0,0,0,0,0,0,0,0,0,0,0,1e-8,0.1,0];
```

这些参数的意义如下：

OPTIONS(1) —— 显示参数（预置值为:0)，当其为 1 时，将显示某些结果；
OPTIONS(2) —— 坐标位置(X)终止标准(预置值为：1e-4)；
OPTIONS(3) —— 目标函数值终止标准(预置值为：1e-4)；
OPTIONS(4) —— 约束条件违背终止标准(预置值为：1e-6)；
OPTIONS(5) —— 算法选择参数：当其为 1 时，表示选择 FMINS 函数或 LEASTSQ 函数；
OPTIONS(6) —— 算法选择参数：

当其为 0 时，表示选择 FMINU 函数中的 Broyden-Fletcher-Golfarb—Shanno 算法；
当其为 1 时，表示选择 FMINU 函数中的 Davidon-Fletcher-Powell 算法；
当其为 2 时，表示选择 FMINU 函数中的 Steepest Descent 算法；

OPTIONS(7) —— 算法选择参数：一维搜索算法(Line Search Algorithm).（预置值为 0）
OPTIONS(8) —— 返回函数值；
OPTIONS(9) —— 如想检查用户提供的梯度值时，可将其设置为 1(预置值为 0)；
OPTIONS(10) —— 返回该优化过程进行计算的总次数；
OPTIONS(11) —— 返回该优化过程对函数梯度进行计算次数；
OPTIONS(12) —— 返回该优化过程对约束条件进行计算次数；
OPTIONS(13) —— 返回该优化过程对等式约束进行计算次数；
OPTIONS(14) —— 对函数进行计算的最大次数(预置值为 100 乘以变量数)；
OPTIONS(15) —— 只用于最优控制的目标达到(goal attainment)过程；
OPTIONS(16) —— 只用于有限差分梯度过程；
OPTIONS(17) —— 只用于有限差分梯度过程(finite difference gradients)；
OPTIONS(18) —— 步长（预置值为 1 或更小值）。

三、模糊 *K*-均值聚类方法应用示例

```
>iris =[
  2.3000   3.2000   4.3000   5.6000   1.0000
  2.4000   3.3000   4.6000   5.1000   1.0000
  2.7000   2.9000   4.6000   5.8000   1.0000
  2.7000   3.2000   4.0000   5.9000   1.0000
  2.1000   3.6000   4.7000   5.2000   1.0000
  3.1000   2.1000   5.8000   4.3000   2.0000
  3.2000   2.5000   5.3000   4.0000   2.0000
  3.7000   2.4000   5.6000   3.9000   2.0000
  3.2000   2.7000   5.9000   4.1000   2.0000
  3.5000   2.8000   5.1000   3.7000   2.0000
  4.1000   2.9000   2.1000   2.8000   3.0000
  4.2000   2.4000   2.7000   2.0000   3.0000
  4.7000   2.1000   3.6000   3.1000   3.0000
  3.9000   3.6000   2.9000   3.2000   3.0000
  4.5000   2.7000   2.8000   3.6000   3.0000];
>class1 = iris(find(iris(:,5)==1), :);   % data for class 1
>class2 = iris(find(iris(:,5)==2), :);   % data for class 2
>class3 = iris(find(iris(:,5)==3), :);   % data for class 3
>data = iris;            % whole data set to be clustered
>data_n = size(data , 1);          % number of data
>expo = 2.0;          % Exponent for U
```

```
>cluster _ n = 3;              % Number of clusters
>max _ iter = 100;             % Max. iteration
>min _ impro = 1e-6;           % Min. improvement
>obj _ fcn = zeros(max _ iter, 1);       % Array for objective function
>digitH = zeros(cluster _ n, 6);         % Array for handles of cluster centers
>U = initfcm(cluster _ n, data _ n);        % Initial fuzzy partition
>[U, center] = stepfcm(data, U, cluster _ n, expo);% Initial cluster centers
>mark = '. ; % Project data to 2-D and plot them
>seq = [1 2; 1 3; 1 4; 2 3; 2 4; 3 4];
>for j = 1:6,
>    x = seq(j, 1); y = seq(j, 2);   % inputs selected for plot
>    subplot(2,3,j);
>    h = plot([class1(:,x) class2(:,x) class3(:,x)], …
>            [class1(:,y) class2(:,y) class3(:,y)], mark);
>     set(h, 'markersize', 8);
>     xlabel(['x' int2str(x)]); ylabel(['x' int2str(y)]);
>for k = 1:cluster _ n, % Initialize centers
>            digitH(k,j) = text(center(k, seq(j,1)),…
>                 center(k,seq(j,2)), int2str(k));
>            set(digitH(k,j), 'erase', 'xor', 'horizon', 'center');
>     end
>end
>for i = 1:max _ iter, % Main loop
>    [U, center, obj _ fcn(i)] = stepfcm(data, U, cluster _ n, expo);
>    fprintf('Iteration count = %d, obj. fcn = %f\n', i, obj _ fcn(i));
>    if i > 1, % check termination condition
>        if abs(obj _ fcn(i) - obj _ fcn(i-1)) < min _ impro, break; end,
>    end
>for j = 1:6,
>        for k = 1:cluster _ n,
>            set(digitH(k,j), 'pos', center(k, seq(j,:)));
>        end
>    end
>    drawnow;
>end
>iter _ n = i.              % Plot of objection function
>figure;                    % New figure
>obj _ fcn(iter _ n+1:max _ iter) = [];% Delete unused elements
>plot(obj _ fcn); axis([1 i min(obj _ fcn) max(obj _ fcn)]);
>title('Objective Function for IRIS Data Clustering');
>xlabel('Numbers of Iterations');
```

四、关于使用 MATLAB 中小波变换工具箱（toolbox）函数的简要说明

1. 未加工小波基函数

小波基函数：morlet 小波函数，mexican hat(mexihat)小波函数。

性质：φ

φ 函数不存在。

— 基函数不正交。

— ϕ 函数(psi)不致密支撑(compactly supported)。

— 重构性质不保险。

可能分析：

— 连续分解。

主要优点：对称，ϕ 函数有显式。

主要困难：没有重构和快速算法。

2. 无限规则小波(Infinitely regular wavelets)

meyer 小波

性质：

— φ 函数存在且分析是正交的。

— ϕ 函数和 φ 函数是无限可导的。

— ϕ 函数和 φ 函数不致密支撑。

可能进行的分析：

— 连续变换

— 可进行离散变换。

主要优点：对称，无限规则。

主要困难：无快速算法。

3. 正交且致密支撑的小波函数

Daubechies 小波函数(dbN)，coiflets 小波函数(coifN)，symlets 小波函数(symN)

一般性质：

— φ 函数存在且分析是正交的。

— ϕ 函数和 φ 函数是致密支撑的。

可能进行的分析：

— 连续变换。

— 可用快速小波变换(FWT)进行离散变换。

主要优点：致密支撑、存在滤波器。

主要困难：规则性不强。

特殊性质：

对于 dbN ：不对称(asymmetry)。

对于 symN ：接近对称。

对于 coifN：接近对称。

4. 双正交且致密支撑的小波函数对

B-样条双正交小波函数(biorNr. Nd)。

性质：

— φ 函数存在且分析是双正交的。

— 进行分解和重构的 ϕ 函数和 φ 函数是数密支撑的。

— 进行重构的 ϕ 函数和 φ 函数有可知规则性。

— 可进行速续变换。

— 可用快速小波变换(FWT)进行离散变换。

主要优点：对滤波器而言是对称的。

主要困难：正交性不存在。

有关 Daubechies 小波函数的信息

家族名： Daubechies

短名： db

水平 N： N 为正整数

例子： db1 or haar，db4，db15

正交性： yes

双正交性： yes

致密支撑： yes

离散小波变换 possible

连续小波变换 possible

支撑宽度： 2N－1

滤波器长度： 2N

对称性： 没有

参考文献：I. Daubechies，Ten lectures on wavelets，CBMS，SIAM，61，1994，194-202.

有关 coiflets 小波函数的信息

家族名： Coiflets

短名： coif

水平 N： N ＝ 1，2，…，5

例子： coif2，coif4

正交性： yes

双正交性： yes

致密支撑： yes

离散小波变换 possible

连续小波变换 possible

支撑宽度： 6N－1

滤波器长度： 6N

对称性： 接近对称

参考文献：I. Daubechies，Ten lectures on wavelets，CBMS，SIAM，61，1994，258-261.

五、MATLAB 中小波工具箱使用示例

例1：一维离散小波变换

1. 装载原始一维信号：

```
〉load leleccum；
〉s = leleccum (1：3920)；
〉ls = length (s)；
```

2. 将此原始一维信号在屏幕上画出来：

```
〉subplot (211)；
〉plot (1：ls，s,'r')；
〉title ('Original signal s.')；
```

3. 对信号用 Daubechies 小波函数（dbN），进行一步分解，并将结果在屏幕上画出：

```
〉[ca1，cd1] = dwt (s,'db1')；…
〉subplot (223)；
〉plot (1：length (ca1)，ca1)；
```

```
》title ('Approx. coef. : ca1');
》subplot (224);
》plot (1: length (cd1), cd1);
》title ('Detail coef. : cd1');
```

4. 对所得分解系数ca1和cd1进行重构，并将结果在屏幕上画出：

```
》a1 = upcoef('a',ca1,'db1',1,ls);
》d1 = upcoef('d',cd1,'db1',1,ls);
》h1 = subplot(311); plot(1:ls,s,'r');
》title('Original signal s.');
》h2 = subplot(312); plot(1:ls,a1);
》title('Approximation : a1.');
》h3 = subplot(313); plot(1:ls,d1);
》title('Detail : d1.');
```

5. 对所得a1和d1相加，并将结果在屏幕上画出；figure(active fig)

```
h1 = subplot(311); plot(1:ls,s,'r');
title('Original signal s.');
h2 = subplot(312); plot(1:ls,a1 + d1);
title('Approximation + detail : a1 + d1.');
set(h1,'Xtick',[],'Xticklabel',[]);
```

6. 用所得系数对信号进行小波逆变换：

```
》a0 = idwt(ca1,cd1,'db1',ls);
h3 = subplot(313); plot(1:ls,a0);
title('Reconstructed signal : a0.');
```

7. 对信号用Daubechies小波函数(dbN)，进行水平3的分解，并将结果在屏幕上画出：

```
》[c,l] = wavedec(s,3,'db1');
h1 = subplot(511); plot(1:ls,s,'r');
title('Original signal s.');
```

8. 用所得小波分解结构[c,l]，来得到水平3的逼近系数，并将结果在屏幕上画出：

```
》ca3 = appcoef(c,l,'db1',3);
》h2 = subplot(5,8,9); plot(1:length(ca3),ca3);
》subplot(511); title('Original signal s and ca3.');
```

9. 用所得小波分解结构[c,l]，来得到水平1、2、3的细节系数，并将结果在屏幕上画出：

```
》cd3 = detcoef(c,l,3);
》cd2 = detcoef(c,l,2);
》cd1 = detcoef(c,l,1);
》subplot(511); title('Original signal s , ca3 and cd3.');
》h3 = subplot(5,8,17); plot(1:length(cd3),cd3);
》pause(2)
》subplot(511); title('Original signal s , ca3, cd3 and cd2.');
》h4 = subplot(5,4,13); plot(1:length(cd2),cd2);
》pause(2);
》subplot(511); title('Original signal s , ca3, cd3, cd2 and cd1.');
》h5 = subplot(5,2,9); plot(1:length(cd1),cd1);
```

10. 用所得小波分解结构[c,l]，来得到水平3的重构逼近系数，并将结果在屏幕上画出：

```
》a3 = wrcoef('a',c,l,'db1',3);
h1 = subplot(511); plot(s,'r');
title('Original sig. app. level 3.');
h2 = subplot(512); plot(1:ls,a3);
```

11. 用所得小波分解结构[c,l]，来得到水平1、2、3的重构细节系数，并将结果在屏幕上画出：

```
》d3 = wrcoef('d',c,l,'db1',3);
》d2 = wrcoef('d',c,l,'db1',2);
》d1 = wrcoef('d',c,l,'db1',1);
subplot(511); title('Original signal and a3 d3 d2 d1.');
h3 = subplot(513); plot(1:ls,d3);
h4 = subplot(514); plot(1:ls,d2);
h5 = subplot(515); plot(1:ls,d1);
```

12. 用所得小波分解结构[c,l]，来重构分析信号：

```
》a0 = waverec(c,l,'db1');
subplot(211); plot(s,'r');
title('Original signal s.');
subplot(212); plot(1:ls,a0);
title('Reconstructed signal a0.');
```

例2：Mallat 塔式算法（FWT）示例

1. 构建原始信号：

```
》s = 2 + kron (ones (1, 8), [1 -1]) + ( (1: 16) .^ 2) /32 + 0.2*randn (1, 16);
》subplot (311); plot (s,'r');
》title ('Original signal s.');
```

2. 对一给定的正交小波函数计算两个分解滤波器：

```
》[Lo_D, Hi_D] = wfilters ('db1','d');
》x_fil = 1: length (Lo_D);
》subplot (323); wdstem (x_fil, Lo_D,'y');
》xlabel ('Lo_D.');
》subplot (324); wdstem (x_fil, Hi_D,'y');
》xlabel ('Hi_D.');
```

3. 检察分解低通滤波器的和及其模：

```
》sm = sum (Lo_D);
》subplot (323); xlabel ( ['Lo_D : sum = ', num2str (sm)]);
》nrm = norm (Lo_D);
》subplot (323); xlabel ( ['Lo_D : norm = ', num2str (nrm)]);
```

4. 用Daubechies小波函数（dbN）计算信号的逼近系数，首先是求信号s和低通分解滤波器的卷积，然后计算它的二分的下采样：

```
》tempo = conv (s, Lo_D);
》subplot (312); plot (tempo);
》title ('s and Lo_D convolution.');
》ca1 = dyaddown (tempo);
》subplot (325); plot (ca1);
》xlabel ('Approx. coef. : ca1.');
》subplot (323); wdstem (x_fil, Lo_D,'y');
```

```
》xlabel ('Lo _ D.');
》subplot (324); wdstem (x _ fil, Hi _ D,'y');
》xlabel ('Hi _ D.');
```

5. 检察分解高通滤波器和与模：

```
》sm = sum (Hi _ D);
》subplot (324); xlabel ( ['Hi _ D : sum = ', num2str (sm)]);
》rm = norm (Hi _ D);
》subplot (324); xlabel ( ['Hi _ D : norm = ', num2str (nrm)]);
```

6. 用 Daubechies 小波函数（dbN）计算信号的细节，首先是求信号 s 和高分解滤波器的卷积，然后计算它的二分的下采样：

```
》empo = conv (s, Hi _ D);
》ubplot (312); plot (tempo);
》itle ('s and Hi _ D convolution.');
》d1 = dyaddown (tempo);
》ubplot (326); plot (cd1);
》label ('Detail coef. : cd1.');
```

7. 也可利用离散小波变换一步计算信号的逼近系数和细节系数：

```
》[1, cd1] = dwt (s,'db1');
》subplot (325); plot (ca1,'g');
》xlabel ('Approx. coef. : ca1.');
》subplot (326); plot (cd1,'g');
》xlabel ('Detail coef. : cd1.');
```

8. 下一步显示怎样用两个重构滤波器来进行重构：

```
》[Lo _ R, Hi _ R] = wfilters ('db1','r');
》subplot (323); wdstem (x _ fil, Lo _ R,'y'); % plot (Lo _ R);
》xlabel ('Lo _ R.');
》subplot (324); wdstem (x _ fil, Hi _ R,'y'); % plot (Hi _ R);
》xlabel ('Hi _ R.');
》subplot (321); plot (ca1,'g');
》title ('Approx. coef. : ca1.');
》subplot (322); plot (cd1,'g');
》title ('Detail coef. : cd1.');
```

9. 重构细节系数：第一步对 cd1进行二分上采样计算，第二步再对其与高通滤波器进行卷积计算，最后对所得结果取其中部16个元素，即进行 wkeep 运算：

```
》tempo == dyadup (cd1);
》subplot (313); plot (tempo);
》xlabel ('Upsampled det. coeff. cd1.');
》tempo = conv (tempo, Hi _ R);
》subplot (313); plot (tempo);
》xlabel ('Upsampled detail coeff. and Hi _ R convolution.');
》d1 = wkeep (tempo, 16);
》subplot (313); plot (d1);
》xlabel ('Reconstructed detail d1.');
```

10. 重构逼近系数：第一步对 ca1进行二分上采样计算，第二步再对其与低通滤波器进行卷积计算，最后

对所得结果取其中部16个元素，即进行 wkeep 运算：

```
》tempo = dyadup (ca1);
》subplot (313); plot (tempo);
》xlabel ('Upsampled approx. coeff. ca1');
》tempo = conv (tempo, Lo_R);
》subplot (313); plot (tempo);
》xlabel ('Upsampled approx coeff. and Lo_R convolution.');
》a1 = wkeep (tempo, 16);
》subplot (313); plot (a1);
》xlabel ('Reconstructed approx. a1.');
```

11. 最后可将逼近系数和细节系数加合，即得重构的原始信号：

```
》a0 = a1 + d1;
》title ('Original signal s.');
》subplot (312); plot (1: 16, a1, 1: 16, d1);
》title ('a1 and d1.');
》h3 = subplot (313); plot (a0);
》title ('Reconstructed signal : a0 = a1 + d1.');
```

12. 前述过程亦可用离散小波逆变换一步完成：

```
》a0 = idwt (ca1, cd1,'db1', 16);
》subplot (313); plot (a0,'g');
》title ('Reconstructed signal a0.');
```

例3：用小波变换对一维信号除噪

1. 设定信噪比和随机噪声函数的种子，产生带有高斯白噪声的原始信号，并将其在屏幕上显示：

```
》snr = 3; init = 2055615866;
》[xref,x] = wnoise(3,11,snr,init);
》subplot(311);
》plot(xref,'r'), axis([1 2048 -10 10]);
》title('Original signal');
》subplot(312);
》plot(x,'g'), axis([1 2048 -10 10]);
》title(['Noisy signal-Signal to noise ratio = ',num2str(fix(snr))]);
```

2. 用水平5的小波 sym8函数，取无标度的自推理变化原则(heursure)的软去噪模式对信号进行去噪处理，并将其在屏幕上显示：

```
》lev = 5;
》xd = wden(x,'heursure','s','one',lev,'sym8');
》subplot(313); plot(xd); axis([1 2048 -10 10])
》title('Denoised signal-heuristic SURE');
```

3. 用水平5的小波 sym8函数，取无标度的 Stein 的无偏风险估计(Stein's Unbiased Risk Estimate)原则(rigrsure)的软去噪模式对信号进行去噪处理，并将其在屏幕上显示：

```
》xd = wden(x,'rigrsure','s','one',lev,'sym8');
》subplot(313); plot(xd); axis([1 2048 -10 10]);
title('Denoised signal-SURE');
```

4. 用水平5的小波 sym8函数，取无标度的阈值为 sqrt(2 * log(length(X)))的原则(sqtwolog)的软去噪模式对信号进行去噪处理，并将其在屏幕上显示：

```
》xd = wden(x,'sqtwolog','s','sln',lev,'sym8');
》subplot(313); plot(xd); axis([1 2048 -10 10])
》title('Denoised signal - Fixed form threshold');
```

5. 用水平5的小波 sym8函数，取无标度的极小极大阈值选择的原则(minimaxi)的软去噪模式对信号进行去噪处理，并将其在屏幕上显示：

```
》xd = wden(x,'minimaxi','s','sln',lev,'sym8');
》subplot(313); plot(xd); axis([1 2048 -10 10])
》title('Denoised signal-Minimax');
```

6. 用水平5的小波 sym8函数，先对信号矢量进行分解，以得到它的分解结构[c,l]，然后再进行除噪处理：

```
》[c,l] = wavedec(x,lev,'sym8');
》xd = wden(c,l,'minimaxi','s','sln',lev,'sym8');
```

7. 装载一个分析信号，选择其中一部分，首先用函数 ddencmp 取得函数 wdencmp 的预置值，继用函数 wdencmp 进行除噪，并将结果在屏幕上显示：

```
》load leleccum; indx = 2600:3100;
》x = leleccum(indx);
》subplot(211);
》plot(indx,x,'r'), title('Original signal')
》[thr,sorh,keepapp] = ddencmp('den','wv',x);
》xd = wdencmp('gbl',x,'db3',2,thr,sorh,keepapp);
》subplot(212);
》plot(indx,xd), title('Denoised signal')
```

8. 另取一例继续试验：

```
》init = 2055615866;
》snr = 5; % signal to noise ratio.
》[xref,x] = wnoise(1,11,snr,init);
》indx = linspace(0,1,length(x));
》subplot(211);
》plot(indx,x,indx,xref)
》title('Noisy and Original signals')
》xlabel(['Signal to noise ratio = ',num2str(fix(snr))])
》xd = wden(x,'heursure','s','one',5,'sym8');
》subplot(212);
》plot(indx,xd,indx,xref), xlabel('Denoised and Original signals')
```

9. 改变信噪比，再取一例继续试验：

```
》snr = 4; % signal to noise ratio.
》[xref,x] = wnoise(2,11,snr,init);
》indx = linspace(0,1,length(x));
》subplot(211);
》plot(indx,x,indx,xref)
》title('Noisy and Original signals')
》xlabel(['Signal to noise ratio = ',num2str(fix(snr))])
》xd = wden(x,'rigrsure','s','one',5,'sym4');
》subplot(212);
》plot(indx,xd,indx,xref), xlabel('Denoised and Original signals')
```

例4：用小波变换对信号进行压缩处理

1. 装载一个分析信号，选择其中一部分，并将其在屏幕上显示：

```
〉load leleccum; indx = 2600:3100;
〉x = leleccum(indx);
〉subplot(211);
〉plot(indx,x,'r'), title('Original signal')
```

2. 用函数 wdencmp 来进行信号压缩，采用固定阈值(35)，并将结果在屏幕上显示：

```
〉thr = 35;
〉[xd,cxd,lxd,perf0,perfl2] = wdencmp('gbl',x,'db3',3,thr,'h',1);
〉subplot(212);
〉plot(indx,xd), title('Compressed signal')
〉xlabel(['2-norm rec. : ',num2str(perfl2,'%5.2f'),...
        ' % - nul cfs: ',num2str(perf0,'%5.2f'), ' %'])
```

3. 装载一图像信号，并将其在屏幕上显示：

```
〉load woman; x = X(100:200,100:200);
〉nbc = size(map,1);
〉colormap(pink(nbc));
〉image(wcodemat(x,nbc)), title('Original image')
```

4. 用函数 wdencmp 来进行图像信号压缩，并将结果在屏幕上显示：

```
〉n = 5; w = 'sym2';
〉[c,l] = wavedec2(x,n,w);
〉thr = 20;
〉[xd,cxd,lxd,perf0,perfl2] = wdencmp('gbl',c,l,w,n,thr,'h',1);
〉image(wcodemat(xd,nbc))
〉title(['Compressed Image. Threshold = ',num2str(thr)])
〉strxlab = strvcat(['2—norm rec. : ',num2str(perfl2,'%5.2f') ' %'],...
        ['nul cfs : ',num2str(perf0,'%5.2f'), ' %']);
〉xlabel(strxlab);
〉axis('square');
```

六、MATLAB 中人工神经网络程序使用示例

例1：

1. 先定义输入矢量 P：

```
〉P = [-3.0 +2.0];
```

2. 定义目标矢量 T：

```
T = [+0.4 +0.8];
```

3. 用 ERRSURF 函数计算用一个神经元计算在可能的权重和偏置范围内的误差，并用 PLOTES 函数画出误差图和等高线图：

```
〉wv = -4:0.4:4;
〉bv = -4:0.4:4;
〉es = errsurf(P,T,wv,bv,'logsig');
〉plotes(wv,bv,es,[60 30]);
```

误差图中的最小值即为最好点。

4. 用 INITFF 函数来对 LOGSIG 神经元的权重和偏置进行初始化：

```
》[w,b] = initff(P,T,'logsig')
》w =
    1.1200
》b =
   -0.9456
```

5. 给出必要参数,用TBP1函数(一层前传误差反传网络)训练人工神经网络:

```
》df = 5;        % Frequency of progress displays (in epochs).
》me = 100;      % Maximum number of epochs to train.
》eg = 0.01;% Sum-squared error goal.
》lr = 2;        % Learning rate.
》  [w,b,ep,tr] = tbp1(w,b,'logsig',P,T,[df me eg lr],wv,bv,es,[60 30]);
```

以下为屏幕显示:

```
TRAINBP: 0/100 epochs, SSE = 0.985712.
TRAINBP: 5/100 epochs, SSE = 0.970767.
TRAINBP: 10/100 epochs, SSE = 0.779827.
TRAINBP: 15/100 epochs, SSE = 0.531896.
TRAINBP: 20/100 epochs, SSE = 0.154784.
TRAINBP: 25/100 epochs, SSE = 0.0438716.
TRAINBP: 29/100 epochs, SSE = 0.00887375.
```

该程序的输出参数为:

w:权重结果;

b:偏置结果;

ep:训练总次数;

tr:误差记录。

用下述绘图函数可看出训练次数与误差的关系:

```
》ploterr(tr,eg);
```

还可用 simuff 函数来验证是否可由任意给定输入而得到目标值:

在本例,给 -3应得到目标值0.4;

给+2.0应得到目标值+0.8。

```
》p = -1.2;
》a = simuff(p,w,b,'logsig')
a =
    0.4881
```

例2:

1. 先定义输入矢量 P:

```
》P = -1:.1:1;
```

2. 定义目标矢量 T:

```
T = [-.9602   -.5770   -.0729    .3771    .6405    .6600    .4609    ...
       .1336   -.2013   -.4344   -.5000   -.3930   -.1647    .0988    ...
       .3072    .3960    .3449    .1816   -.0312   -.2189   -.3201 ];
```

3. 用作图程序来观察一下输入与输出的非线性关系:

```
plot(P,T,'+');
title('Training Vectors');
xlabel('Input Vector P');
```

ylabel('Target Vector T');

4. 设计网络：

本例用一二层网络，TANSIG 转换函数和 purelin 转换函数为其神经元，S1为其隐含层神经元个数：

```
》S1 = 5；
```

5. 用 INITFF 函数来对网络的权重和偏置进行初始化：

```
》[w1,b1,w2,b2] = initff(P,S1,'tansig',T,'purelin');
```

6. 给出必要参数，用 trainbp 函数来训练人工神经网络：

```
》df = 10; % Frequency of progress displays (in epochs).
》me = 8000; % Maximum number of epochs to train.
》eg = 0.02; % Sum-squared error goal.
》lr = 0.01; % Learning rate.
》tp = [df me eg lr];
》[w1,b1,w2,b2,ep,tr] = trainbp(w1,b1,'tansig',w2,b2,'purelin',P,T,tp);
```

以下为屏幕显示：

```
TRAINBP: 0/8000 epochs, SSE = 7.95707.
TRAINBP: 10/8000 epochs, SSE = 4.29622.
TRAINBP: 20/8000 epochs, SSE = 3.78964.
TRAINBP: 30/8000 epochs, SSE = 3.36207.
TRAINBP: 40/8000 epochs, SSE = 2.99139.
TRAINBP: 50/8000 epochs, SSE = 2.66459.
TRAINBP: 60/8000 epochs, SSE = 2.37366.
TRAINBP: 70/8000 epochs, SSE = 2.11346.
TRAINBP: 80/8000 epochs, SSE = 1.88035.
TRAINBP: 90/8000 epochs, SSE = 1.67151.
TRAINBP: 100/8000 epochs, SSE = 1.48453.
TRAINBP: 110/8000 epochs, SSE = 1.31726.
TRAINBP: 120/8000 epochs, SSE = 1.16777.
TRAINBP: 130/8000 epochs, SSE = 1.03428.
TRAINBP: 140/8000 epochs, SSE = 0.915206.
TRAINBP: 150/8000 epochs, SSE = 0.809095.
TRAINBP: 160/8000 epochs, SSE = 0.714641.
TRAINBP: 170/8000 epochs, SSE = 0.630665.
TRAINBP: 180/8000 epochs, SSE = 0.556099.
TRAINBP: 190/8000 epochs, SSE = 0.489974.
TRAINBP: 200/8000 epochs, SSE = 0.431414.
TRAINBP: 210/8000 epochs, SSE = 0.379622.
TRAINBP: 220/8000 epochs, SSE = 0.333877.
TRAINBP: 230/8000 epochs, SSE = 0.293524.
TRAINBP: 240/8000 epochs, SSE = 0.257972.
TRAINBP: 250/8000 epochs, SSE = 0.226687.
TRAINBP: 260/8000 epochs, SSE = 0.199185.
TRAINBP: 270/8000 epochs, SSE = 0.175034.
TRAINBP: 280/8000 epochs, SSE = 0.153844.
TRAINBP: 290/8000 epochs, SSE = 0.135269.
```

TRAINBP: 300/8000 epochs, SSE = 0.118996.

TRAINBP: 310/8000 epochs, SSE = 0.10475.

TRAINBP: 320/8000 epochs, SSE = 0.092284.

TRAINBP: 330/8000 epochs, SSE = 0.0813809.

TRAINBP: 340/8000 epochs, SSE = 0.0718476.

TRAINBP: 350/8000 epochs, SSE = 0.0635139.

TRAINBP: 360/8000 epochs, SSE = 0.0562296.

TRAINBP: 370/8000 epochs, SSE = 0.0498628.

TRAINBP: 380/8000 epochs, SSE = 0.0442973.

TRAINBP: 390/8000 epochs, SSE = 0.0394315.

TRAINBP: 400/8000 epochs, SSE = 0.0351761.

TRAINBP: 410/8000 epochs, SSE = 0.031453.

TRAINBP: 420/8000 epochs, SSE = 0.0281939.

TRAINBP: 430/8000 epochs, SSE = 0.0253393.

TRAINBP: 440/8000 epochs, SSE = 0.0228372.

TRAINBP: 450/8000 epochs, SSE = 0.020642.

TRAINBP: 454/8000 epochs, SSE = 0.0198409.

该程序的输出参数为:

w1——第一层权重结果;

b1——第一层偏置结果;

w2——第二层权重结果;

b2——第二层偏置结果;

ep——训练总次数;

tr——误差记录。

7. 用下述绘图函数可看出训练次数与误差的关系:

〉ploterr(tr,eg);

还可用 simuff 函数来验证是否可由任意给定输入而得到目标值:

在本例,给 0.5应得到目标值0.3506;

〉p = 0.5;

〉a = simuff(p,w1,b1,'tansig',w2,b2,'purelin')

a =

0.3506

例3: 一个拟合度不够(UNDERFITTING)的示例

1. 先定义输入矢量 P:

〉P = −1:.1:1;

2. 定义目标矢量 T:

T = [−.9602 −.5770 −.0729 .3771 .6405 .6600 .4609 …

.1336 −.2013 −.4344 −.5000 −.3930 −.1647 .0988 …

.3072 .3960 .3449 .1816 −.0312 −.2189 −.3201];

3. 用作图程序来观察一下输入与输出的非线性关系:

plot(P,T,'+');

title('Training Vectors');

xlabel('Input Vector P');

ylabel('Target Vector T');

4. 设计网络：

本例用一二层网络，TANSIG 转换函数和 purelin 转换函数为其神经元，S1为其隐含层神经元个数（神经元个数太少时，容易导致拟合度不够）：

```
》S1 = 2;
```

5. 用 INITFF 函数来对网络的权重和偏置进行初始化：

```
》[w1,b1,w2,b2] = initff(P,S1,'tansig',T,'purelin');
```

6. 给出必要参数，用 trainlm 函数来训练人工神经网络：

```
》df = 10; % Frequency of progress displays (in epochs).
》me = 200 ; % Maximum number of epochs to train.
》eg = 0.02; % Sum-squared error goal.
》tp = [df me eg ];
》[w1,b1,w2,b2,ep,tr] = trainlm(w1,b1,'tansig',w2,b2,'purelin',P,T,tp);
```

以下为屏幕显示：

```
TRAINLM: 0/200 epochs, mu = 0.001, SSE = 33.3699.
TRAINLM: 10/200 epochs, mu = 0.1, SSE = 2.15034.
TRAINLM: 20/200 epochs, mu = 0.001, SSE = 0.729464.
TRAINLM: 30/200 epochs, mu = 0.001, SSE = 0.724132.
TRAINLM: 40/200 epochs, mu = 0.001, SSE = 0.722526.
TRAINLM: 50/200 epochs, mu = 0.0001, SSE = 0.719605.
TRAINLM: 60/200 epochs, mu = 0.0001, SSE = 0.718858.
TRAINLM: 70/200 epochs, mu = 0.0001, SSE = 0.718515.
TRAINLM: 80/200 epochs, mu = 0.0001, SSE = 0.718307.
TRAINLM: 90/200 epochs, mu = 1e-005, SSE = 0.717681.
TRAINLM: 100/200 epochs, mu = 1e-005, SSE = 0.71749.
TRAINLM: 110/200 epochs, mu = 1e-005, SSE = 0.717395.
TRAINLM: 120/200 epochs, mu = 1e-005, SSE = 0.717335.
TRAINLM: 130/200 epochs, mu = 1e-005, SSE = 0.717293.
TRAINLM: 140/200 epochs, mu = 1e-005, SSE = 0.717261.
TRAINLM: 150/200 epochs, mu = 1e-006, SSE = 0.717138.
TRAINLM: 160/200 epochs, mu = 1e-006, SSE = 0.717083.
TRAINLM: 170/200 epochs, mu = 1e-006, SSE = 0.717055.
TRAINLM: 180/200 epochs, mu = 1e-006, SSE = 0.717037.
TRAINLM: 190/200 epochs, mu = 1e-006, SSE = 0.717024.
TRAINLM: 200/200 epochs, mu = 1e-006, SSE = 0.717014.
TRAINLM: Network error did not reach the error goal.
  Further training may be necessary, or try different
  initial weights and biases and/or more hidden neurons.
```

此结果说明用 TRAINLM 训练只用2个隐蔽层结点的神经网络，不能得到满意结果。

例4：一个过拟合（OVERFITTING）的示例

1. 先定义输入矢量 P 和目标矢量 T：

```
》P = -1:.1:1;
》T = [-.9602 -.5770 -.0729 .3771 .6405 .6600 .4609 …
      .1336 -.2013 -.4344 -.5000 -.3930 -.1647 .0988 …
```

.3072 .3960 .3449 .1816 −.0312 −.2189 −.3201];

2. 用作图程序来观察一下输入与输出的非线性关系：

plot(P,T,'-');

title('Training Vectors');

xlabel('Input Vector P');

ylabel('Target Vector T');

3. 设计网络：

本例用一二层网络，TANSIG 转换函数和 purelin 转换函数为其神经元，S1为其隐含层神经元个数（隐含层神经元个数太多时，容易导致过拟合）：

> S1 = 40;

4. 用 INITFF 函数来对网络的权重和偏置进行初始化：

> [w1,b1,w2,b2] = initff(P,S1,'tansig',T,'purelin');

5. 给出必要参数，用 trainlm 函数来训练人工神经网络：

> df = 10; % Frequency of progress displays (in epochs).

> me =8000 ; % Maximum number of epochs to train.

> eg = 0.02; % Sum-squared error goal.

> tp = [df me eg];

> [w1,b1,w2,b2,ep,tr] = trainlm(w1,b1,'tansig',w2,b2,'purelin',P,T,tp);

以下为屏幕显示：

TRAINLM: 0/8000 epochs, mu = 0.001, SSE = 73.6058.

TRAINLM: 4/8000 epochs, mu = 0.001, SSE = 0.0186811.

SSE 的结果已经达到目标。

6. 用下述绘图函数可看出训练次数与误差的关系：

> ploterr(tr,eg);

7. 用 simuff 函数来验证目标曲线的拟合情况，并将结果在屏幕上显示：

> P2 = -1:0.01:1;

> A = simuff(P2,W1,b1,'tansig',W2,b2,'purelin');

> plot(P,T,'+',P2,A,'m');

> title('Testing the Network');

> xlabel('Input Vector P');

> ylabel('Target Vector T');

此结果说明 TRAINLM 训练由40个隐蔽层结点的神经网络，很容易产生过拟合。

例5：一个用人工神经网络来进行英文字母识别的例子：

1. 首先装载数据：函数 prprob 将给出两个矩阵，一个矩阵 alphabet 为一：

> [alphabet,targets] = prprob;

> [R,Q] = size(alphabet);

函数 prprob 将给出两个矩阵，其中 alphabet 为一(35×26)阶矩阵，它的每一列表示一个字母的位图(bit map)；targets 为一(26×26)阶的单位矩阵，它的每一列表示一个字母的标识矢量，如字母 A 的标识为一个第一个元素为1，其余元素为0的矢量；而字母 Z 的标识则为一个第26个元素为1，其余元素为0的矢量。

用下列程序可清楚地看出每一个字母的位图

> for k=1:26

> for i=1:7

> for j=1:5

> a(i,j,k)=alphabet((i-1)*5+j,k);

```
> end
> end
> end
```

2. 设计网络：

本例用一二层网络，仍用 logsig 转换函数为其神经元，S1为其隐含层神经元个数：

```
> S1 = 10;
```

3. 用 INITFF 函数来对网络的权重和偏置进行初始化：

```
> [W1,b1,W2,b2] = initff(alphabet,S1,'logsig',targets,'logsig');
> W2 = W2 * 0.01;
> b2 = b2 * 0.01;
```

4. 给出必要参数，用 trainbpx 函数来训练人工神经网络：

```
> df = 20;        % Frequency of progress displays (in epochs).
> me = 5000;       % Maximum number of epochs to train.
> eg = 0.1;       % Sum-squared error goal.
> mc = 0.95;       % Momentum constant.
> tp = [df me eg NaN NaN NaN mc];
> P = alphabet;
> T = targets;
> [W1,b1,W2,b2,ep,tr] = trainbpx(W1,b1,'logsig',W2,b2,'logsig',P,T,tp);
```

以下为屏幕显示：

```
TRAINBPX: 0/5000 epochs, lr = 0.01, SSE = 168.808.
TRAINBPX: 20/5000 epochs, lr = 0.026533, SSE = 26.6316.
TRAINBPX: 40/5000 epochs, lr = 0.0500319, SSE = 25.0037.
TRAINBPX: 60/5000 epochs, lr = 0.0500319, SSE = 25.1697.
TRAINBPX: 80/5000 epochs, lr = 0.0814967, SSE = 25.173.
TRAINBPX: 100/5000 epochs, lr = 0.216235, SSE = 25.0259.
TRAINBPX: 120/5000 epochs, lr = 0.573736, SSE = 24.3849.
TRAINBPX: 140/5000 epochs, lr = 1.52229, SSE = 22.1595.
TRAINBPX: 160/5000 epochs, lr = 4.03909, SSE = 15.051.
TRAINBPX: 180/5000 epochs, lr = 10.7169, SSE = 5.05021.
TRAINBPX: 200/5000 epochs, lr = 28.4352, SSE = 0.863142.
TRAINBPX: 208/5000 epochs, lr = 42.0117, SSE = 0.0984008.
```

达到目标，训练结束。

5. 训练含噪声的数据：

先将原训练结果转入 W1n 、b1n 、W2n 、b2n，以备训练含噪声的数据用，

```
> W1n = W1; b1n = b1; W2n = W2; b2n = b2;
```

给出相应参数：

```
> me = 300;        % Maximum number of epochs to train.
> eg = 0.6;       % Sum-squared error goal.
> tp = [df me eg];
```

进行五次含随机噪声的数据训练：

```
> T = [targets targets targets targets];
> for pass = 1:5
>    fprintf('Pass = %2.0f\n',pass);
```

```
》 P = [alphabet, alphabet, …
》       (alphabet + randn(R,Q) * 0.1), …
》       (alphabet + randn(R,Q) * 0.2)];
》 [W1n,b1n,W2n,b2n,ep,tr] =
   trainbpx(W1n,b1n,'logsig',W2n,b2n,'logsig',P,T,tp);
》 end
```

结果显示：

```
fprintf('Pass = %2.0f\n',pass);
Pass =  1
TRAINBPX: 0/300 epochs, lr = 0.01, SSE = 4.31546.
TRAINBPX: 20/300 epochs, lr = 0.026533, SSE = 2.39066.
TRAINBPX: 40/300 epochs, lr = 0.0703999, SSE = 1.43271.
TRAINBPX: 60/300 epochs, lr = 0.186792, SSE = 0.961293.
TRAINBPX: 80/300 epochs, lr = 0.495614, SSE = 0.695128.
TRAINBPX: 89/300 epochs, lr = 0.768861, SSE = 0.592093.
end
  fprintf( Pass = %2.0f\n',pass);
Pass = 2
  [W1n,b1n,W2n,b2n,ep,tr] = trainbpx(W1n,b1n,'logsig',W2n,b2n,'logsig',P,T,tp);
TRAINBPX: 0/300 epochs, lr = 0.01, SSE = 3.22597.
TRAINBPX: 20/300 epochs, lr = 0.026533, SSE = 1.89788.
TRAINBPX: 40/300 epochs, lr = 0.0703999, SSE = 1.33045.
TRAINBPX: 60/300 epochs, lr = 0.186792, SSE = 0.976347.
TRAINBPX: 80/300 epochs, lr = 0.495614, SSE = 0.725668.
TRAINBPX: 92/300 epochs, lr = 0.890052, SSE = 0.590583.
end
  fprintf('Pass = %2.0f\n',pass);
Pass = 3
  P = [alphabet, alphabet, …
  [W1n,b1n,W2n,b2n,ep,tr] = trainbpx(W1n,b1n,'logsig',W2n,b2n,'logsig',P,T,tp);
TRAINBPX: 0/300 epochs, lr = 0.01, SSE = 5.07201.
TRAINBPX: 20/300 epochs, lr = 0.026533, SSE = 3.03126.
TRAINBPX: 40/300 epochs, lr = 0.0703999, SSE = 1.59253.
TRAINBPX: 60/300 epochs, lr = 0.186792, SSE = 0.968169.
TRAINBPX: 80/300 epochs, lr = 0.495614, SSE = 0.653241.
TRAINBPX: 85/300 epochs, lr = 0.632544, SSE = 0.597847.
end
  fprintf('Pass = %2.0f\n',pass);
Pass = 4
  P = [alphabet, alphabet, …
  [W1n,b1n,W2n,b2n,ep,tr] = trainbpx(W1n,b1n,'logsig',W2n,b2n,'logsig',P,T,tp);
TRAINBPX: 0/300 epochs, lr = 0.01, SSE = 4.01717.
TRAINBPX: 20/300 epochs, lr = 0.026533, SSE = 2.94552.
TRAINBPX: 40/300 epochs, lr = 0.0703999, SSE = 1.72363.
```

```
TRAINBPX: 60/300 epochs, lr = 0.186792, SSE = 1.04955.
TRAINBPX: 80/300 epochs, lr = 0.495614, SSE = 0.695784.
TRAINBPX: 87/300 epochs, lr = 0.697379, SSE = 0.59675.
end
  fprintf('Pass = %2.0f\n',pass);
Pass = 5
  P = [alphabet, alphabet, ...
  [W1n,b1n,W2n,b2n,ep,tr] = trainbpx(W1n,b1n,'logsig',W2n,b2n,'logsig',P,T,tp);
TRAINBPX: 0/300 epochs, lr = 0.01, SSE = 2.72573.
TRAINBPX: 20/300 epochs, lr = 0.026533, SSE  1.69701.
TRAINBPX: 40/300 epochs, lr = 0.0703999, SSE = 1.00911.
TRAINBPX: 60/300 epochs, lr = 0.186792, SSE = 0.694741.
TRAINBPX: 69/300 epochs, lr = 0.289775, SSE = 0.598531.
end
```

训练结束。

6. 用含噪声的数据训练所得来验证该网络是否能正确识别无噪声的原始数据给出必要参数进行检验：

```
> df = 5;          % Frequency of progress displays (in epochs).
> me = 500;        % Maximum number of epochs to train.
> eg = 0.1;        % Sum-squared error goal.
> tp = [df me eg];
> P = alphabet;
> T = targets;
> [W1n,b1n,W2n,b2n,ep,tr] = trainbpx(W1n,b1n,'logsig',W2n,b2n,'logsig',P,T,tp);
```

结果显示：

```
TRAINBPX: 0/500 epochs, lr = 0.01, SSE = 0.0642643.
```

毋需训练即可得到正确结果。

7. 比较经含噪声训练所得结果(network2)与不含噪声训练所得结果(network1)是否不同给出检验参数：

```
> noise_range = 0:.05:.5;
> max_test = 100;
> network1 = [ ];
> network2 = [ ];
> T = targets;
```

执行检验：

```
> for noiselevel = noise_range
>   fprintf('Testing networks with noise level of %.2f.\n',noiselevel);
> Testing networks with noise level of 0.00.
>   errors1 = 0;
>   errors2 = 0;
>   for i=1:max_test
>     P = alphabet + randn(35,26) * noiselevel;
>     A = simuff(P,W1,b1,'logsig',W2,b2,'logsig');
>     AA = compet(A);
>     errors1 = errors1 + sum(sum(abs(AA-T)))/2;          % 检验 NETWORK 1
```

```
》    An = simuff(P,W1n,b1n,'logsig',W2n,b2n,'logsig');
》    AAn = compet(An);
》    errors2 = errors2 + sum(sum(abs(AAn - T)))/2;
》  echo off
```

以下为屏幕显示结果：

```
Testing networks with noise level of 0.05.
Testing networks with noise level of 0.10.
Testing networks with noise level of 0.15.
Testing networks with noise level of 0.20.
Testing networks with noise level of 0.25.
Testing networks with noise level of 0.30.
Testing networks with noise level of 0.35.
Testing networks with noise level of 0.40.
Testing networks with noise level of 0.45.
Testing networks with noise level of 0.50.
```

用图将检验结果显示出来：

```
》clf
》plot(noise_range,network1 * 100,'--',noise_range,network2 * 100);
》title('Percentage of Recognition Errors');
》xlabel('Noise Level');
》ylabel('Network 1 _ _ Network 2 \_\_\_');
```

从图可清楚看出，经含噪声训练所得结果(network2)与不含噪声训练所得结果(network1)。

七、遗传算法程序使用说明

1. 实数遗传算法程序中的变量说明

psize	染色体个体数目(即需调优的参数的个数)
pmutate	变异率
mutatec	变异步长
cyc	循环次数
step	步长
popu	染色体
popuerr	与染色体(即需调优的参数)对应的误差
errflow	误差流
popufit	染色体适应性
son 1	交叉子代
son2	变异子代
sonerr	与子代对应的误差
sonfit	子代的适应性
parent	父代
offspr	通过竞争产生的新子代
matesite	交叉点
fitsum	适应性的总和
site	变异点

2. 实数遗传算法程序所包含的子程序

mate.m 交叉函数

compet.m 竞争函数

fitness.m 适应性函数

mutate.m 变异函数

select.m 选择性交叉函数

3. 构建一个遗传优化的主程序

因此遗传优化函数是为求极大设计的，故如要求极小时，需对目标函数取负值，本例就是如此。

```
  rga.m
  psize=20;pmutate=0.3;mutatec=0.001;
  cyc=0;step=ones(1,psize) * 0.001;
  popu=rand(3,psize)-0.2;
  popuerr=lmsob(popu,x1,y1);
  errflow=[];
while cyc<500
     cyc=cyc+1;
     errflow=[errflow,popuerr(1)];
     popufit=fitness(popuerr);
     parent=select(popufit);
     son1=mate(popu(:,parent));
     son2=mutate(son1,pmutate,mutatec);
     son=[son1,son2];
     sonerr=lmsob(son,x1,y1);
     sonfit=fitness(sonerr);
     popu=[popu,son];
     popufit=[popufit,sonfit];
     popuerr=[popuerr,sonerr];
     offspr=compet(popufit,psize);
     popu=popu(:,offspr);
     popuerr=popuerr(offspr);
end
plot(errflow)
disp('The optimal individual is')
popu(:,1)
```

4. 结果检验

```
>plot(x1 * popu(:,1))
> hold
>plot(y1,'r')
```

说明：因遗传算法为一随机优化算法，其参数的选择是很重要的，如 变异率（pmutate）、变异步长（mutatec）、循环次数（cyc）、步长（step）等，不正确的参数选择将可能得到的是局部最优，更一般说来，遗传算法还是依赖所需解决问题的特殊性的，所以，在使用上述程序时应加以注意。

附　　录

一、化学计量学中常用名词中英文对照表

A

ASTM 红外光谱索引	ASTM infrared spectral index

B

Bayes 判别法	Bayes discrimination
半因子设计法	half-Fraction factorial design
半正定	positive semidefinite
饱和设计	saturated design
饱和 *D*-最优设计	saturated *D*-optimal design
崩溃点	breakdown point
比特	bit
变尺度法	variant scale method
变量	variable
变换	transformation
编码	coding
编码信息量	coding information
编码效率	coding efficiency，CE
变步长通用模拟退火法	variable step-size generalized simulated annealing
标准加入迭代目标转换因子分析法	standard addition iterative target transformation factor analysis
标准化	normalization
病态体系	Ill-conditioned system
波动噪声	fluctuation or flicker noise
波数	wave number，WN
波长	wave length，λ
不等性方差	heteroscedastic variance
部分因子设计	fractional factorial design
不确定度	uncertainty

C

采样理论	sampling theory
采样常数	sampling constants
残差正态图	residual normal plot
残差双线性分解法	residual bilinearization
产生子	generator
产生式规则表达	generating rule expressing

乘幂法	power method
乘同余法	multiple congruential method
窗口 Fourier 变换	windowed Fourier Transform，WET
窗口移动平均法	window moving average method
窗口移动中位数平滑	window moving median smoothing
窗口因子分析法	window factor analysis
重复中位数法	repeated median method
纯分析信号	net analytical signal，NAS
纯分析物种的秩	net analyte rank，NAR
尺度	scaling
尺度函数	scaling function
尺度滤波	scaling filter
尺度向量	scaling vector
次轴	secondary axes
抽检特性曲线	operating characteristic curve
抽出误差	extracted Error，XE
粗糙惩罚平滑法	roughness penalty smoothing method
簇项	cluster terms
簇类的独立软模式	soft independent modeling of class analogy，SIMCA

D

D-最优设计	*D*-optimal design
代表性采样	representative sampling
单位矩阵	identity matrix
单纯形优化	simplex optimization
单纯形试验设计法	simplex experiment design
单纯形格子点	simplex lattice point
单纯形格子设计	simplex lattice design
单纯形重心设计	simplex gravity center design
单个中位数法	single median method
等价定理	equivalence law
等性方差	homoscedastic variance
得分矩阵	scores matrix
迭代目标转换因子分析法	iterative target transformation factor analysis
定义关系	defining relation
定义相反	defining contrast
定量结构活性关系研究	quantitative structure-activity relationship，QSAR
定量结构性质关系研究	quantitative structure-property relationship，QSPR
定量测定限	quantitative detection limit
独立性	independence
短时 Fourier 变换	short time Fourier transform，STFT
对称矩阵	symmetric matrix
多步法	multistep method
多分辨	multiresolution

多分辨分析	multiresolution analysis
多分辨信号分解	multiresolution signal decomposition, MRSD
多环芳烃	polycycle aromatic hydrocarbons, PAH
多元分辨	multivariate resolution
多元校正	multivariate calibration
多元线性回归方法	multiple linear regression

E

二次饱和 *D*-最优设计	second-order saturated *D*-optimal design

F

F-分布	*F*-isher distributions
F-检验	*F*-test
Fisher 判别法	Fisher discrimination
Fourier 变换	Fourier transformation
范围标度化	range scaling
方差分析	analysis of variance
方差平方和法	variance squares method
方阵	square matrix
非均匀性	heterogeneity
非奇异矩阵	non-singular matrix
非线性分析体系	non-linear analytical systems
非线性迭代偏最小二乘算法	nonlinear iterative partial least squares, NIPALS
分辨率	resolution
分布	distribution
分层采样	stratified sampling
分隔	segregation
分块矩阵	partitioned matrices
分析化学品质因数	figures of merit of analytical chemistry
分析信号	analytical signal
分析信息理论	analytical information theory
分析物-内标参考相关法	analyte-internal reference correlated method, AIRCM
分子结构编辑模块	molecular structure editor
分子连接性指数	molecular connectivity index
分子拓扑指数	molecular topological index
分子图	molecular graph
分子识别数	molecular identification number

G

感知器	perceptron
杠杆点	leverage point
刚性微分方程	stiff differential equations
供信能力	information power
共轭效应	conjugated effect

共轭梯度	conjugate gradient
共现	confounded
Gy 理论	Gy's theory
固定尺寸移动窗口渐进因子分析法	fix-sized moving window evolving factor analysis
光谱多重性效益	multichannel advantage in spectroscopy
广度优先搜索	breadth-first search
广义内标法	generalized internal reference method
广义逆	generalized inverse
广义秩消失因子分析法	generalized rank annihilation factor analysis
惯量因子	momentum factor

H

Hadamard 变换	Hadamard transformation
Hammett σ 常数	Hammett σ constant
行列式	determinant
行满秩	row full rank
行矢量	row vector
好格子点法	good lattice point
活性函数	activation function
荷载矩阵	loadings matrix
互信息	mutual information
化学模式识别	chemical pattern recognition
化学专家系统	chemical expert system
化学知识系统	chemical knowledge-based system
化学振荡反应	chemical oscillating reaction
回归距离和	regressive distance sums
回归顶点点价	regressive vertex degrees
回溯	backtrack
混合同余法	mixed congruential method
混料试验设计	mixture experiment design

J

极大似然估计	maximum likelihood estimator
计算机数字模拟法	computer numerical simulation
计量抽样检验	amounting sample test
计数抽样检验	numbering sample test
基于数论方法的序贯优化法	sequential number-theoretic optimization method
加权最小二乘回归法	weighted least squares regression
交叉校验法	cross-validation
交叉效应	interactive effect
交互效应表	interaction effect tables
检测限	detection limit
简易分类算法	simple classification algorithm, SIMCA
间接校正方法	indirect calibration methods

渐进因子分析法 evolving factor analysis
结构图 structural or constitutional graph
结构解析 elucidation of structure
结构主干 structure backbone
经典诊断方法 classic diagnostic methods
局部顶点不变数 local vertex invariants，LOVT's
局部优化算法 local optimal method
局部曲线拟合法 local curve fitting method
局部因子分析 local factor analysis
句法模式识别技术 syntactic pattern recognition
聚类分析法 clustering analysis
距离判别法 distance discrimination method
矩阵校正方法 matrix calibration method
矩阵的秩 rank of matrix
矩阵分辨方法 matrix resolution method
矩阵的迹 trace of matrix
矩阵 matrix
基于全局寻优的聚类法 clustering method based on global optimization
兼并性或退化性 degeneracy
卷积 convolution
均匀分布的伪随机数 quasi-random number of uniform distribution
均匀设计 uniform experimental design
均匀设计表 uniform design table
均匀性 homogeneity
均匀性准则 criterion for uniformity

K

K-矩阵法 *K* matrix method
K-均值聚类法 *K*-means clustering method
K-最近邻法 *K*-nearest neighbor classification method，*K*-NN
Kalman 滤波 Kalman filtering
Kalman 增益 Kalman gain
颗粒性质因子 particle property factor
可疑度 equivocation
可接受质量水平 acceptable quality level
可变类平均法 variable class average method
可变法 variable method
可靠性分析 reliability analysis
控制-推理器模块 controller-reasoner
快速 Fourier 变换 fast fourier transform，FFT

L

拉丁方 latin square
类平均法 class average method

离差矩阵	dispersion matrix
离散 Fourier 变换	discrete Fourier transform，DFT
离散小波变换	discrete wavelet transformation
粒状物质	granular material
立体效应常数	steric effect parameters
连接函数	connectivity function
链项	chain terms
列矢量	column vector
列满秩	column full rank
邻接矩阵	adjacency matrix
临界水平值	transition level
岭回归法	ridge regression
岭迹图	ridge trace plot
灵敏度	sensitivity
龙格-库塔法	Range-Kutta method
逻辑表达	logical expressing
路径项	path terms
路径/簇项	path/cluster terms

M

M 估计法	*M*-estimation
马氏距离	Mahalanobis distance
帽子矩阵	hat matrix
盲目搜索	blind search
幂等矩阵	idempotent matrix
模	norm
模糊聚类法	fuzzy clustering method
模拟退火法	simulated annealing method
模式空间	pattern space
模式空间的距离	distance in pattern space
Monte Carlo 模拟	Monte Carlo simulation
Moore-Penrose 逆	Moore-Penrose inverse
目标	object
目标引导	goal-directed

N

奈特	nat
内标法	internal reference method，IRM
内部相关	inner relation
内积	inner product
拟水平均匀设计	pseudo-level design
逆向推理	inverse reasoning
逆矩阵	inverse matrix
牛顿法	Newton method

O

欧氏距离	Euclidean distance

P

P-矩阵法	P-matrix method
判别分析法	discrimination method
判决面	discrimination plane
批不合格品百分数的允许界限	lot tolerance percent defective
偏差	discrepancy；deviation
偏置	biases
偏最小二乘法	partial least squares
偏最小二乘的投影判别法	projection discrimination based on PLS
平滑方法	Smooth method
平移	shift or translation
Powell 法	Powell method

Q

奇异矩阵	singular matrix
奇异值分解法	single value decomposition，SVD
启发式分类	heuristic classification
启发式搜索	heuristic search
前传网络误差反传算法	back-propagation，BP
取代基	substituent
取代基电子效应常数	substituent electronic effect constant
全局优化算法	global optimal method
全局因子分析法	global factor analysis

R

Randic 分支指数	Randic branch index
人工神经网络	artificial neural network
人工神经网络判别法	discrimination based on artificial neural network
人工智能	artificial intelligence
冗余信息	redundancy

S

SIMCA 分类法	SIMCA classification
s 型非线性函数	sigmoid nonlinearity function
熵	entropy
散度	divergence
神经纤维	nerve fibres
神经元	neuron
深度优先搜索	depth-first search
识别率	recognition ratio

矢量	vector
矢量校正方法	vector calibration method
释放因子	liberation factor
四分之一因子设计法	quarter-fraction factorial design
势函数判别法	potential function discrimination
剩余标准偏差	residual standard deviation, RSD
双线性矩阵	bilinear matrix
数据库管理模块	librarian
数据推动	data driven
数据预处理	data preprocessing
数据专家模块	data expert
数字模拟方法	digital simulation
搜索树	searching tree
苏尔兹分子拓扑指数	Schultz molecular topological indices
顺序秩分析方法	sequential rank analysis
随机变量的数值特征	numerical feature of random variable
随机采样	random sampling
随机事件	random events
随机变量	random variable
所有顶点的均方距离指标	mean-square distance between all vertexes

T

t-检验	t-test
t-分布	t-distribution
塔式算法	pyramid algorithm
台劳级数法	taylor series method
梯度法	gradient method
条件熵	conditional entropy
特征抽取方法	feature extraction
特征空间	feature space
特征矢量	eigenvector
特征投影的模式识别方法	pattern recognition by latent projection
特征值	eigenvalue
统一匀线性多元分析	statistical isolinear multiple component analysis, SIMCA
同名或别名	alias
通用标准加入法	generalized standard addition method
投影算法	projection algorithm
投影寻踪法	projection pursuit
投影指标	projection index
投影寻踪回归	projection pursuit regression
投影寻踪分类	projection pursuit classification
投影寻踪密度估计	projection pursuit density estimation
投影旋转因子分析法	projection rotation factor analysis
投影阵	projection matrix

突触	dendrirtes
图像处理	image processing
椭球多变量修剪法	ellipsoidal multivariate trimming, MVT

W

Wiener 拓扑数	Wiener topological index
伪随机数	quasi-random number
稳健校正	robust calibration
稳健主成分回归	robust principal component regression
稳健偏最小二乘法	robust partial least squares
稳健诊断方法	robust diagnostic method
稳健距离	robust distance
稳健主成分分析	robust principal component analysis, RPCA
稳健奇异值分解法	robust single value decomposition, RSVD
无监督的模式识别	unsupervised pattern recognition

X

系统采样	systematic sampling
系统聚类法	hierarchical clustering method
线性比率内标法	line ratio internal reference method, LRIRM
线性不相关	linearly independent
线性相关	linearly dependent
线性自由能	linear free-energy
线性自由能关系	linear free-energy relationship
线性图	linear graph
线性学习机	linear learning machine
线性判别函数法	linear discrimination function method
线性组合设计	linear combinatorial design
限制发生器	constrained generator, CONGEN
相似性	similarity
响应值	response
小波函数	wavelet function
小波多分辨变换	wavelet multiresolution transformation
小波域	wavelet domain
兴奋剂	agonist
信道容量	channel capacity
信息量	information content
信息效率	information profitability
信噪比	signal-to-noise ratio, SNR
选择性	selectivity
学生分布统计量	student Distribution statistics

Y

样本	sample

样本对象	object
一次饱和 D-最优设计	first-order saturated D-optimal design
一点设计	single point design
一分为二法	one divided in two method
一阶谓词逻辑	first order predicate logic，FOPL
遗传算法	genetic algorithm
因子分析	factor analysis
因子分析的误差理论	error theory of factor analysis
因子设计	factorial design
因子设计分辨率	resolution of factorial design
隐蔽层	hidden layer
隐氢图	hydrogen-depleted graph
有机反应性相关分析	correlation analysis of organic reactivity
有监督的模式识别	supervised pattern recognition
有监督的学习	supervised learning
有偏估计方法	biased estimating method
有限总体校正	finite population correction
诱导效应	inducing effect
与/或图	AND/OR-graph
阈值	threshold
预测率	prediction ratio
语义网络表达	semantic net expressing
原子质量单位	atom mass unit，AMU
约束背景双线性分解	constrained background bilinearization

Z

杂散噪声	shot noise
张量	tensor
张量分辨方法	tensor rosolution method
张量校正理论	tensor calibration theory
直观推导式演进特征投影法	heuristic evolving latent projections
直接校正方法	direct calibration method
植入误差	imbedded error
知识表达	knowledge expressin
秩消失因子分析法	rank annihilation factor analysis
正定	positive definite
正交设计	orthogonal design
正交设计表	orthogonal arrays
正交投影阵	orthogonal projection matrix
正交投影分辨法	orthogonal projection resolution
正态分布的伪随机数	quasi-random numbers of normal distribution
正态分布	normal distrihutions
正交矩阵	orthogonal matrix
正交性	orthogonality

振荡	oscillation
诊断方法	diagnostic methods
"钟乳石图"	stalactite plot
真实误差	real error，RE
中间距离法	medium distance method
中位数	median
重心法	gravity center method
主成分回归法	principal component regression
主成分投影判别法	projection discrimination based on PCA
主效应	main effect
主轴	primary axes
逐步判别分析法	stepwise discrimination method
转置	transpose
子窗口因子分析法	subwindow factor analysis
子空间	subspace
自标度化	autoscaling
自适应 Kalman 滤波法	adaptive Kalman filtering
自适应线性元	adalnei
自模式曲线分辨法	self-modeling curve resolution
阻滞剂	blocker
组成因子	composition factor
最长距离法	maximum distance method
最短距离法	minimum distance method
最小采样数目	minimum number of samples
最小体积椭球估计	minimum volume ellipsoid estimates，MVE
最小中位平方法	least median of squares
最小修剪平方法	least trimmed squares
最小一乘法	least absolute residuals
最小生成树	minimun spanning tree
最优试验设计	optimal design
坐标轮换法	coordinate changing method

二、符号说明

$\boldsymbol{A}$	吸光度
$\boldsymbol{A}^{+}$	矩阵 $\boldsymbol{A}$ 的广义逆或称 Moore-Penrose 逆
$\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$，…	矩阵
$\boldsymbol{A}^{t}$，$\boldsymbol{B}^{t}$，$\boldsymbol{C}^{t}$，…	转置矩阵
$\boldsymbol{a}$，$\boldsymbol{b}$，$\boldsymbol{c}$，…	小写黑体字母为列矢量
$\boldsymbol{a}^{t}$，$\boldsymbol{b}^{t}$，$\boldsymbol{c}^{t}$，…	行矢量
$\boldsymbol{A}_{p\times q}$	（$p\times q$）阶矩阵 $\boldsymbol{A}$，即矩阵 $\boldsymbol{A}$ 具有 p 行 q 列
$\boldsymbol{I}$	单位矩阵
$\boldsymbol{1}$	所有分量皆为1的列矢量
$\boldsymbol{0}$	所有分量皆为0的列矢量
det（$\boldsymbol{A}$）	$\boldsymbol{A}$ 的行列式